

UNIVERSIDADE FEDERAL DO PARANÁ

ADOLFO LUIZ NORILLER LEITE DA SILVA

DEIVIDI ALEXANDRE CAVARZAN

FELIPE CESÁRIO BAESSO

LEANDRO REZENDE LEMOS

JOGO MYCITY PARA REDES SOCIAIS

CURITIBA

2012

ADOLFO LUIZ NORILLER LEITE DA SILVA

DEIVIDI ALEXANDRE CAVARZAN

FELIPE CESÁRIO BAESSO

LEANDRO REZENDE LEMOS

JOGO MYCITY PARA REDES SOCIAIS

Trabalho apresentado à disciplina de Trabalho de Conclusão de Curso, do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor Escola Técnica da Universidade Federal do Paraná.

Orientador: Professor Mestre Rafael Romualdo Wandressen

CURITIBA

2012

SUMÁRIO

RESUMO.....	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Contextualização	12
1.2 Justificativa do Projeto.....	12
1.3 Objetivos.....	13
1.3.1 Objetivo Geral	13
1.3.2 Objetivos Específicos	13
1.4 Organização do Projeto	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 Jogos e Redes Sociais	14
2.2 Indicadores	16
2.3 Metodologias Ágeis e SCRUM	19
2.4 Perspectiva Isométrica	22
2.5 Game Design.....	24
2.6 Facebook.....	24
2.6.1 Ingresso na aplicação.....	25
2.7 Considerações Finais	26
3 METODOLOGIA.....	27
3.1 Plano de Atividades	27
3.2 Responsabilidade	27
3.3 Scrum	28
3.3.1 Stories	28
3.3.2 Burn down	32
3.4 Plano de Riscos.....	34
3.5 Plano de Custos	36

3.6	Materiais	37
3.6.1	Linguagem de programação PHP	37
3.6.2	Doctrine	37
3.6.3	AMFPHP	37
3.6.4	Adobe Flex	38
3.6.5	MySQL	38
3.6.6	As3lib	38
3.6.7	API Facebook.....	38
3.6.8	Scrum	38
4	APRESENTAÇÃO DO SOFTWARE	39
4.1	Game Design.....	39
4.1.1	Objetivo	39
4.1.2	Cenário.....	40
4.1.3	Construções	40
4.1.4	Estado Inicial	41
4.1.5	Passagem de Tempo (Turnos).....	42
4.1.6	Socialização	42
4.1.7	Referência	42
4.2	Telas do Software.....	42
5	CONSIDERAÇÕES FINAIS.....	52
5.1	Melhorias	52
5.2	Continuidade.....	53
	REFERÊNCIAS BIBLIOGRÁFICAS	54
	APÊNDICE – Documentação do software	57
	Apêndice 1 - Gráficos de burn down.....	57
	Apêndice 1.1 Burn Down – Sprint 1	57
	Apêndice 1.2 Burn Down – Sprint 2.....	58

Apêndice 1.3 Burn Down – Sprint 3	59
Apêndice 1.4 Burn Down – Sprint 4	60
Apêndice 2 - Diagrama de WBS	61
Apêndice 3 - Fluxo de Gantt	62
Apêndice 3.1 Fluxo de Gantt para os Quatro Sprints do Projeto	62
Apêndice 3.2 Fluxo de Gantt para o Sprint 1	63
Apêndice 3.3 Fluxo de Gantt para o Sprint 2	64
Apêndice 3.4 Fluxo de Gantt para o Sprint 3	65
Apêndice 3.5 Fluxo de Gantt para o Sprint 4	66
Apêndice 3.6 Fluxo de Gantt Detalhado para os Quatro Sprints	67
Apêndice 4 - Diagrama de Classes	68
Apêndice 5 - Diagrama de Casos de Usos	69
Apêndice 6 - Descrição dos Casos de Uso.....	69
Apêndice 6.1 UC01 - Acessar aplicação.....	69
Apêndice 6.2 UC02 – Verificar cidade	70
Apêndice 6.3 UC03 - Criar nova cidade.....	70
Apêndice 6.4 UC04 - Carregar cidade	72
Apêndice 6.5 UC05 - Construir	72
Apêndice 6.6 UC06 - Demolir	76
Apêndice 6.7 UC07 - Alocar operário	77
Apêndice 6.8 UC08 - Convidar amigos.....	79
Apêndice 6.9 UC09 - Verificar status.....	80
Apêndice 7 - Diagrama Entidade Relacionamento	82
Apêndice 8 - Análise por Pontos por Caso de Uso	83
Apêndice 9 - Diagramas de Atividades	84
Apêndice 10 - Diagrama de Estados	90

LISTA DE SIGLAS

DSDM – Dynamic Software Development Method

OCDE – Organização para a Cooperação e Desenvolvimento Econômico

ONU – Organização das Nações Unidas

SNG – Social Network Games – Jogos para Redes Sociais.

WBS – Work Breakdown Structure

XP – Extreme programming.

LISTA DE FIGURAS

FIGURA 1: Exemplo de SNG voltado para Facebook. FONTE: Os autores	15
FIGURA 2: Exemplo da divisão das estórias em um <i>product backlog</i> e um sprint. O tamanho de cada retângulo representa a importância da estória e a chave representa as estórias que a equipe conseguirá finalizar durante o sprint.....	21
FIGURA 3: Exemplo de um gráfico de burndown. A reta em azul representa a quantidade de horas restantes para cada dia. A linha em vermelho representa o andamento da equipe.	22
FIGURA 4: Representação gráfica das perspectivas isométricas. (a) Perspectiva cônica; (b) Perspectiva cavaleira; (c) Perspectiva isométrica.	23
FIGURA 5: Representação da malha isométrica e de um objeto isométrico e seus eixos cartesianos para o jogo MyCity.....	24
FIGURA 6: Imagem ilustrativa da ferramenta utilizada.....	28
FIGURA 7: Ilustração da Pirâmide de Maslow mostrando a prioridade das necessidades.....	39
FIGURA 8: Tela inicial do jogo MyCity.	43
FIGURA 9: Painel superior do jogo MyCity com os atributos de satisfação da população.	44
FIGURA 10: Painel de ações inferior do jogo MyCity.....	45
FIGURA 11: Painel de ações expandido com a opção de Construir selecionada. 1. Abas de tipos de construções; 2. Botões de avançar ou voltar nas opções de cada tipo.....	45
FIGURA 12: Janela de atributos de uma construção, ativada ao selecionar uma construção específica.	46
FIGURA 13: Visualização de permissão para construção. A – Construção não permitida, pois está sobreposta com a rua; B – Construção permitida....	47
FIGURA 14: Andamento de uma construção. A – Barra com a quantidade de ciclos realizados em relação ao total para completar a construção; B – Opção para adicionar novos operários na construção.	48
FIGURA 15: Janela de informações de uma construção finalizada.....	48

FIGURA 16: Modo de demolição de construções. A – Ao passar o cursor sobre uma construção que será removida; B – Após pressionar o botão do mouse para demolir.	49
FIGURA 17: Interligação de ruas. A – Ruas isoladas; B – Ruas conectadas automaticamente.	49
FIGURA 18: Painel de amigos e convites do Facebook.	50
FIGURA 19: Painel de convites do Facebook para adição de amigos no jogo MyCity.	51
FIGURA 20: Painel de Configurações do jogo MyCity.	51
FIGURA 21: Gráfico de Burn Down para o Sprint 1	57
FIGURA 22: Gráfico de Burn Down para o Sprint 2	58
FIGURA 23: Gráfico de Burn Down para o Sprint 3	59
FIGURA 24: Gráfico de Burn Down para o Sprint 4	60
FIGURA 25: Diagrama de WBS para o jogo MyCity.	61
FIGURA 26: Gráfico de Gantt para os quatro sprints do jogo MyCity.	62
FIGURA 27: Gráfico de Gantt para o Sprint 1.	63
FIGURA 28: Gráfico de Gantt para o Sprint 2.	64
FIGURA 29: Gráfico de Gantt para o Sprint 3.	65
FIGURA 30: Gráfico de Gantt para o Sprint 4.	66
FIGURA 31: Gráfico de Gantt detalhado para os quatro Sprints realizados.	67
FIGURA 32: Diagrama de Classes.	68
FIGURA 33: Diagrama de Casos de Uso para o jogo MyCity.	69
FIGURA 34: Diagrama do Modelo de Entidade Relacionamento para o jogo MyCity.	82
FIGURA 35: Análise por Pontos por Caso de Uso.	83
FIGURA 36: Diagrama de atividades para acessar a aplicação.	84
FIGURA 37: Diagrama de atividades para alocar um operário.	85
FIGURA 38: Diagrama de atividades para construir um item.	86
FIGURA 39: Diagrama de atividades para convidar amigos do Facebook.	87
FIGURA 40: Diagrama de atividades para demolir uma construção.	88
FIGURA 41: Diagrama de atividades para verificar a cidade durante o acesso do jogador.	89
FIGURA 42: Diagrama de estado que ocorre durante uma construção.	90
FIGURA 43: Diagrama descrevendo o acesso à aplicação.	91

FIGURA 44: Diagrama ilustrando o processo para se adicionar uma construção.....	92
FIGURA 45: Diagrama de sequencia descrevendo o processo de listagem de amigos.....	93

LISTA DE TABELAS

TABELA 1: Definição das <i>stories</i> do jogo social MyCity.	29
TABELA 2: Plano de riscos para o jogo MyCity. Primeira condição.	34
TABELA 3: Plano de riscos para o jogo MyCity. Segunda condição.	35
TABELA 4: Plano de riscos para o jogo MyCity. Terceira condição.	36
TABELA 5: Plano de riscos para o jogo MyCity. Quarta condição.....	37

RESUMO

Este trabalho apresenta o desenvolvimento do jogo MyCity para redes sociais, que consiste na criação e manutenção de uma cidade em um ambiente de multi jogadores. A rede social Facebook foi escolhida como ponto focal deste trabalho por possuir um amplo e crescente número de usuários em escala global, permitir uma difusão do jogo de maneira mais ampla e possibilitar uma maior interatividade do jogador com seus amigos. Para isso, o jogo explora este ambiente interativo dentro da rede social, e desafia o jogador a construir uma cidade sustentável e com diversas possibilidades estruturais. Foram criados diversos setores de construção e infra-estrutura, que podem influenciar a cidade virtual por meio de índices de satisfação da população, sustentabilidade e economia. A utilização de dinheiro fictício dentro do jogo permite que o jogador adquira quantias dinheiro para a construção dos itens, potencializando o crescimento de sua cidade, que ocorrerá de uma maneira mais rápida e com itens de valor agregado mais alto, permitindo que em um menor espaço, o jogador possua uma maior produção, seja ela financeira ou de características que satisfaçam a população. Para o desenvolvimento deste projeto foi escolhida a metodologia ágil Scrum, que proporciona uma divisão de tarefas por histórias, auxiliando na diminuição dos prazos de análise, codificação e testes, bem como a divisão de tarefas entre os integrantes da equipe. Dentre as tecnologias escolhidas para a codificação da aplicação foram utilizadas as linguagens de programação PHP, AS3 e Flex, aliadas a *frameworks* específicos para o armazenamento de dados na base de dados, comunicação entre as linguagens, comunicação com a rede social, arquitetura da aplicação e para a confecção da interface do jogo.

Palavras-chaves: rede social, jogo, sustentabilidade

ABSTRACT

This document shows the development of the MyCity game for social networks, which consists on creation and maintenance of a city in a multiplayer environment. The social network Facebook was chosen as the focal point to the development because this network has a large and growing number of users on a global scale, which allow a widely diffusion of the game, allowing for the player a broader and more interactive environment with your friends in social networking. The game explores the interactive environment within the social network and challenges the player to build a sustainable city with various structural possibilities. Have created various types of constructions and infrastructure which way influence the population grow, sustainability and economic environment. The fictitious money used in the game allows the player to get sums of money for the construction items enhancing the growth of the city, which occurs faster and provides items with higher added value, allowing the player to have in small areas a better production, meaning it more profit or attributes that satisfy the population. To develop this project was chosen Scrum agile methodology that provides a division of tasks between stories, assisting in reducing the length of analysis, coding and testing phases, and the division of tasks between team members. Among the technologies chosen for the coding of the application were used programming languages as PHP, AS3 and Flex, coupled with specific frameworks for storing data in the database, communication between languages, communication with social networking, application architecture and for making the game interface.

Key-words: Social Network, Game, Sustainable

1 INTRODUÇÃO

Na última década, a internet acompanhou a popularização das redes sociais, estruturas sociais onde pessoas podem se conectar e compartilhar valores e objetivos. Como uma rede virtual, não se existem barreiras físicas entre os indivíduos, permitindo que as únicas limitações sejam suas próprias identidades e interesses (FIELDS e COTTON, 2011).

A relação humana através do ambiente virtual, fez com que as redes sociais tivessem rápida aceitação e popularização, fazendo com que a quantidade de usuários da principal rede social brasileira, o Facebook, superasse 30 milhões de usuários únicos em dezembro de 2011 (CAMPI, 2011).

1.1 Contextualização

A grande quantidade de pessoas dessas redes chamou a atenção do mercado de jogos digitais, proporcionando o surgimento de uma nova categoria, os Jogos Sociais. Diferentes dos jogos convencionais onde os usuários não se relacionam, os jogos digitais se utilizam da característica fundamental das redes sociais: a troca de informação.

O projeto MyCity busca atender a esse mercado através de um simulador social, voltado para rede social Facebook, onde o usuário se coloca no papel do governante de uma cidade, administrando obras em busca da maior qualidade de vida para sua população. Como desafio, o jogador deve ser capaz de controlar seus recursos, balancear seus ganhos e despesas, decidir como alocar sua mão de obra e selecionar quais construções devem ser utilizadas.

Explorando a principal característica dos jogos sociais, os jogadores poderão interagir entre si de duas formas:

1. Convidar vizinhos: os jogadores terão a possibilidade de convidar amigos das redes sociais para se relacionar dentro do jogo, criando um ciclo de contatos.
2. Compartilhar conquistas: ao atingir certos objetivos, os jogadores recebem opções para publicar mensagens em seus perfis anunciando seus feitos.

1.2 Justificativa do Projeto

O projeto surgiu a partir da ideia de aproveitar o crescente mercado de jogos sociais e criar um produto direcionado a esse público. Segundo estatísticas sobre esse mercado, 53% dos usuários do Facebook jogam games sociais e 20% deles já

pagaram para obter benefícios dentro dos jogos (DE PAULA, 2010). Esse tema foi proposto em 2010, quando simuladores de urbanização ainda não haviam se popularizado. Ainda que no início de 2011 uns dos principais jogos do gênero já tinha alcançado a marca de 100 milhões de jogadores(LEMOS, 2011)optou-se por manter o projeto devido a diferenças significativas no estilo dos jogos.

O mercado atual de jogos sociais cresceu para atender aos milhões de usuários que nunca haviam jogado antes. Em oposição a esse cenário o jogo descrito neste documento tem o objetivo de captar os jogadores mais experientes que almejam por jogos desafiadores e que praticamente ficaram fora do foco das grandes fabricantes na área de jogos sociais

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um jogo online de construção de cidades para redes sociais utilizando metodologias ágeis.

1.3.2 Objetivos Específicos

1. Utilização da metodologia Scrum como base para a análise e acompanhamento do projeto.
2. Desenvolvimento do jogo MyCity através das linguagens de programação PHP e Adobe Flex.
3. Aplicação do MyCity na rede social Facebook.

1.4 Organização do Projeto

Este trabalho é composto por cinco capítulos, incluindo este que é a sua introdução.O segundo capítulo discute os principais conceitos e histórico das redes sociais, descrevendo os aspectos que foram estudados para o desenvolvimento desse projeto. O terceiro capítulo apresenta a metodologia utilizada e o planejamento. Ainda, descreve as tecnologias e bibliotecas utilizadas.O quarto capítulo engloba o planejamento inicial do jogo e a apresentação da sua interface final, servindo como documentação de uso do mesmo.O quinto capítulo aborda a observação do grupo acerca dos desafios encontrados e a reflexão a respeito das técnicas escolhidas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será discutido o cenário em que o projeto será implementado, sua teoria baseada em fatores sociais básicos, a metodologia de desenvolvimento e técnica de apresentação visual aplicada assim como uma introdução sobre o conceito de game design.

2.1 Jogos e Redes Sociais

Os Serviços de Redes Sociais Online, ou Redes Sociais, tiveram um crescimento exponencial no final da última década, alterando o comportamento social e econômico de grande parte da sociedade atual. O crescimento das redes sociais foi acompanhado pelas indústrias de jogos e entretenimento que voltaram diversos produtos, essencialmente jogos casuais, para atender a demanda crescente de usuários que possuíam interesse nesses jogos, ocasionando uma revolução no mercado mundial de jogos com uma alta taxa de lucratividade para a indústria (SHIN,2011).

Em poucos anos os Jogos para Redes Sociais – SNG (Social Network Games) passaram a representar a área mais popular entre os jogos com milhões de usuários jogando periodicamente (SHIN,2011). Entre eles, os jogos que utilizam os contatos dos jogadores para aumentar a interatividade são os que mais amplamente cresceram e ocupam presença na maioria das Redes Sociais. Nesse contexto é possível destacar o grande mercado de trabalho existente para desenvolvedores de jogos.



FIGURA 1: Exemplo de SNG voltado para Facebook. FONTE: Os autores

Os SNG basicamente dispõem de regras e interações, geralmente interligadas a outros jogadores, que permitem que o usuário tenha um grande envolvimento com o jogo. Ele não só escolhe qual jogo quer jogar, mas também com quem ele deseja interagir (JACKSON e WATTS, 2010). Diferentemente dos jogos nos quais o jogador atua sozinho, os jogos sociais buscam o compartilhamento das ações e desempenho do usuário com outros amigos, assim como a colaboração entre os usuários, tornando cada amigo uma fonte de recurso para determinadas ações de um jogo.

Ao contrário de jogos tradicionais, onde os gráficos, som e ambiente são fatores muitas vezes determinantes no sucesso e aceitação do jogo, os SNG tendem a possuir uma maior simplicidade nestes atributos. Além de facilitar o acesso de

qualquer dispositivo ou velocidade de conexão, é possível alcançar usuários de vários níveis de conhecimento computacional, e diminuir consideravelmente os custos com o desenvolvimento. Por exemplo, um jogo tradicional pode custar entre 3 e 4 milhões de reais para ser desenvolvido. Já um jogo social pode custar entre 200 e 300 mil reais (SHIN,2011). A maioria dos jogos sociais possui uma parcela de monetização dentro do jogo, onde jogadores que realizam um investimento adquirem vantagens. O retorno financeiro também pode ser obtido através de anúncios de publicidade dentro do jogo ou na página em que o jogo é executado (FIELDS e COTTON, 2011).

2.2 Indicadores

Abstraindo a administração de uma cidade real, o foco do jogo foi o gerenciamento dos recursos de modo a garantir uma melhor qualidade de vida a população da cidade. Para medir o nível de qualidade de vida é utilizada a idéia de satisfação da população, sendo que esta é obtida conforme os indicadores de desenvolvimento da cidade sejam atingidos, ou seja, pelo resultado das ações do governante da cidade, que deverá gerir as construções de prédios que suportem as necessidades da população, a cidade deverá se aproximar de um grau de satisfação ideal.

Estes indicadores são elementos que dizem respeito as necessidades básicas de qualquer ser humano e de alguns fatores que contribuem para um crescimento do caráter humano. Foram selecionados os indicadores: saúde, segurança e educação como indicadores básicos de qualidade de vida, uma vez que segundo pesquisa Ibope (VENCESLAU, 2010), estas são as principais demandas da população brasileira. Para fatores de crescimento humano foram estipulados como indicadores: lazer, estética e sustentabilidade, fatores que influenciam tanto a condição de vida como a satisfação pessoal do indivíduo.

Cada indicador possui uma serie de atributos que poderiam nos basear para criar o nosso próprio critério de desenvolvimento/qualidade de vida da cidade, mas iremos nos ater a características bastante significativas e que podem de certa forma, simplificar o conceito de indicador:

Saúde: basicamente quando pensamos em saúde, uma das primeiras imagens que temos é a do médico. Um número insuficiente de médicos compromete todo o sistema de saúde e conseqüentemente a população. Segundo recomendação

da OMS, a proporção mínima para que um país não peque no atendimento de saúde ele deve possuir uma taxa 23 médicos (enfermeiros e parteiras também estão inclusos nestes números) para 10000 habitantes, ou seja, 1 médico para aproximadamente 435 habitantes (WHO, 2011). Construções públicas como postos de saúde e hospitais possuem um certo número de profissionais da saúde, e conseqüentemente este número de profissionais poderá atender um certo número de habitantes sem ser sobrecarregado.

Segurança: em nossa sociedade a ordem é assegurada pelos profissionais da segurança, e dentre eles os mais significativos e importantes são os policiais. Para as Nações Unidas. Segundo fontes da ONU (PPCC, 2011), o número de policiais não deve ser inferior a 222 policiais por 100000 habitantes, ou seja, um policial não deveria ser responsável por mais de 450 habitantes (SINGH, 2008). Delegacias e postos de vigilância funcionam por causa do trabalho de policiais, gerando um certo efetivo policial para uma região ou cidade, e este efetivo pode dar suporte de forma eficiente a uma certa parcela da população.

Educação: é um dos principais índices de desenvolvimento humano. Muitos países desenvolvidos têm um sistema educacional bem estruturado e a figura do professor tem grande importância na formação acadêmica dos alunos. Porém muitas vezes, em certos países e regiões há uma falta de professores o que diminui drasticamente a cobertura do sistema educacional e a motivação dos alunos e professores destas regiões, o que prejudica o desenvolvimento nestas áreas. A título de comparação, as nações que compõe o grupo OCDE (Organização para a Cooperação e Desenvolvimento Econômico) (OCDE, 2011), um grupo de 33 países dos quais estão entre os maiores IDH no mundo, e que juntos detêm mais que metade de toda a riqueza do mundo, a média do número professores por habitante, no ensino primário, está em 5 professores para 1000 habitante(1/200) (PETC, 2011) e no ensino secundário a taxa média é de 7 a cada 1000(1/143) (SETC, 2011) .

Sustentabilidade: é um tópico que está em voga ultimamente, e que faz com que queiramos incentivar as pessoas a pensar na forma que elas tratam o ambiente em que vivem. Uma boa forma de incentivar a sustentabilidade é trazendo a importância que as áreas verdes trazem para as cidades. Além do lazer que proporcionam, as áreas verdes ajudam na arborização de centros urbanos, ajudando a diminuir a temperatura e conseqüentemente diminuindo a existência de ilhas de calor (GUZZO, 2011). As áreas verdes também ajudam na melhoria da qualidade do

ar e dos problemas causados pelo acúmulo de água no solo. Segundo a ONU, a área verde ideal seria de 12 m² para cada habitante. (CAPORUSSO, 2008).

Lazer: o lazer traz as pessoas muitos benefícios, sejam eles psicológicos, sociais e físicos. As pessoas tendem a ser mais saudáveis e bem relacionadas conforme tem acesso ao lazer (TEIXEIRA, 2008). Levando em conta o tamanho e capacidade destas instalações, bem como o interesse das pessoas em frequentar estes estabelecimentos iremos nos basear na relação de grandes shoppings para habitantes nos EUA. Este país que oferece uma grande diversidade de entretenimento tem como proporção para seus grandes shoppings e centros de exposições uma relação de 1 para 6000 habitantes (SMTNS, 2011).

Estética: assim como o lazer, a estética urbana também traz benefícios de vida para dos habitantes de uma cidade. Cidades com maiores pontos de interesse como monumentos, parques e prédios arquitetônicos atraem grande número de turistas (PMC, 2011)

Emprego: Como um dos principais motivadores da sociedade, o emprego faz com que um país capitalista funcione, já que sem trabalhadores, a economia e a situação da população entrariam em colapso. Para sabermos a taxa de empregabilidade ou então a taxa de desempregados devemos entender certas questões sobre como é classificada a população quando se trata de emprego. Primeiramente ela é dividida em população em idade ativa e em idade inativo, ou seja, é o número de pessoas que estariam teoricamente em idade apta para trabalhar, sendo que no Brasil, a idade mínima é de 10 anos. Segundo levantamento do IBGE em 2008 a população em idade ativa totalizava 160,6 milhões de pessoas. Destas pessoas em idade ativa podemos separá-las em economicamente ativas e não economicamente ativas. Ainda em 2008, 99,5 milhões de pessoas eram classificadas como economicamente ativas, ou seja, 62% da população em idade ativa. Destas pessoas economicamente ativas (pessoas que estão trabalhando ou que estão a procura de trabalho) podemos encontrar as taxa de empregabilidade e desemprego. Pessoas ditas ocupadas são aquelas que estão trabalhando (ativamente, ou em período de férias) e representam o total de pessoas empregadas, que no Brasil se encontrava em 92,4 milhões. Já as pessoas desocupadas são aquelas que não possuem emprego, mas que estão a procura, ditas desempregados, atingiram a marca de 7,1% da população economicamente ativa. (PNAD, 2008).

2.3 Metodologias Ágeis e SCRUM

As metodologias ágeis foram uma das principais inovações no desenvolvimento de sistema nos últimos anos, tornando-se uma das metodologias mais utilizadas no desenvolvimento de softwares. Entre as mais utilizadas, se destacam as metodologias SCRUM, Extreme Programming (XP), Lean Software Development, Feature-driven Development e DSDM (VLAANDEREN 2011).

O XP é focado nas boas práticas de programação e consiste em doze práticas. As praticas são baseadas no Jogo de Planejamento, pequenos *releases*, design simples, testes, refatoração, programação em pares, integração contínua, jornada de 40 horas semanais, tarefas coletivas, clientes on-site, e padrões de codificação. A segunda revisão do XP, denominada como XP2 possui outros atributos.

O crescimento das novas formas de análise e desenvolvimento ágeis tem forte ligação com a labilidade do processo, em que alterações nos projetos são menos impactantes, o projeto tem uma maior ênfase no desenvolvimento e não em documentações extensas e existe uma maior participação do cliente durante a execução do projeto, evitando a limitação nas funcionalidades que foram descritas no contrato do projeto (MANIFESTO 2001). O aumento da complexidade dos projetos, do número de Stakeholders que participam da definição do produto e dos menores períodos de tempo para a entrega do produto contribuiu para a adoção das metodologias ágeis no mercado de desenvolvimento de softwares (VLAANDEREN, 2011).

A aplicação das metodologias ágeis em empresas é tida como relativamente fácil, embora existam indícios de sua aplicação em grandes empresas possa ser desfavorecida em relação a pequenas empresas. Os benefícios mais comuns nestas aplicações tem como resultado uma maior colaboração do cliente no projeto, processos para controles de erros, estimação de horas para desenvolvimento de tarefas, aprendizado de desenvolvedores em duplas – Utilizado na metodologia XP – e foco centralizado nas atividades atuais (DYBA e DINGSOYR,2008). A qualidade do código-fonte escrito também tende a aumentar com a implementação dessas metodologias.

O SCRUM foi proposto em 1995 por Ken Schwaber (MANIFESTO,2001). O princípio do SCRUM é baseado na premissa de que muitos dos processos que são executados no desenvolvimento de um software nem sempre estão previstos no

escopo do projeto e que o projeto precisa ter flexibilidade para suportar alterações e adição de novos requisitos durante o seu desenvolvimento. Para isso são definidas apenas duas partes no projeto: a primeira e a última fase.

O software é desenvolvido através de uma série de atividades flexíveis chamadas de *sprints*. Para cada *sprints* as atividades são descritas em seu início e não existem adições de novas tarefas durante a sua execução, permitindo que a entrega do produto seja feita com uma maior chance de sucesso e aceitação do cliente, mesmo em casos em que ocorrem grandes mudanças nos requisitos ou no projeto devido a fatores financeiros, temporais ou de mão-de-obra (VLAANDEREN, 2011). Ao avaliar que uma tarefa que foi adicionada ao *sprint* não será desenvolvida na atividade atual, é possível registrar essas tarefas em um backlog, onde posteriormente, poderá ser adicionada em um *sprint* posterior (DYBA e DINGSOYR, 2008). Entre as vantagens dos *sprints*, é a possibilidade da incorporação de novas funcionalidades do sistema ao seu início, aumentando a flexibilidade e diminuindo o impacto no produto e prazos de entrega.

No *product backlog* de um projeto realizado em Scrum são definidas os requisitos e funcionalidade dos sistemas, classificados como *stories*. Em cada história descrita, são definidos alguns itens, como um identificador da história, um nome que irá descrever a história e que auxiliará para distinguir ela das demais, a importância relativa da história que, quanto maior é a pontuação definida, maior é a sua importância e a estimativa inicial de horas em que a tarefa levará para ser executada.

A definição de um *Sprint* é sempre uma tarefa crítica, pois é neste momento em que serão repassadas para a equipe as informações necessárias para o seu desenvolvimento e as histórias são criadas. Normalmente é recomendado que o *product owner* esteja presente no momento desta definição acompanhando as histórias e as estimativas previstas para o desenvolvimento do *Sprint*. Isto ocorre, pois pode ocorrer a mudança de importância das histórias, por exemplo, uma história que irá precisar de mais tempo para o desenvolvimento do que o *product owner* imaginava. Isso pode fazer com que a importância desta história seja reavaliada e classificada novamente. Após este ponto, o *sprint* pode ser definido (Figura 2) e as histórias deste *sprint* dividida entre os membros da equipe, de acordo com a competência ou familiaridade com o seu escopo. Após a etapa de elaboração são executadas pequenas reuniões diárias que ocorrem para acompanhar o andamento

das estórias do *Sprint* atual, e também é delegada à equipe a escolha do melhor horário para esta discussão.

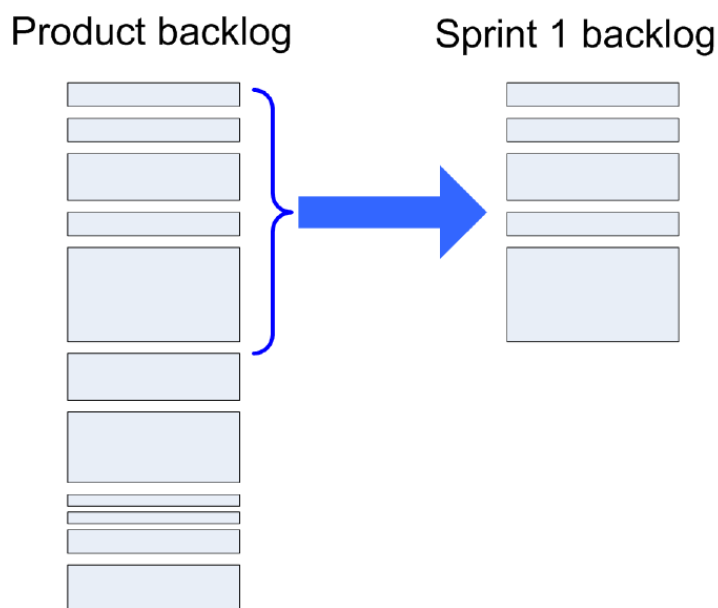


FIGURA 2: Exemplo da divisão das estórias em um *product backlog* e um sprint. O tamanho de cada retângulo representa a importância da estória e a chave representa as estórias que a equipe conseguirá finalizar durante o sprint.

As estórias não devem ser muito grandes e nem muito pequenas. A divisão de estórias em tamanhos muito pequenos acaba gerando um excesso de micro tarefas, dificultando o seu gerenciamento. Ao contrário, estórias grandes correm um grande risco de serem entregues parcialmente, impactando nos demais *sprints*. O ideal é a unificação de estórias de impacto menor e, para estórias maiores, sua divisão em pedaços menores. A divisão ou união das estórias é feita através da discussão com a equipe, envolvendo principalmente o *product owner* e o integrante que irá desenvolver a tarefa, tornando o processo sempre maleável.

Nas reuniões diárias definidas pela equipe é avaliado o andamento das tarefas divididas para cada integrante. Assim é possível a geração de um gráfico que representa o andamento em tempo real das atividades, denominado *burndown*. Este gráfico consiste na equação linear entre as horas restantes *versus* dias totais do *sprint* (Figura 3).

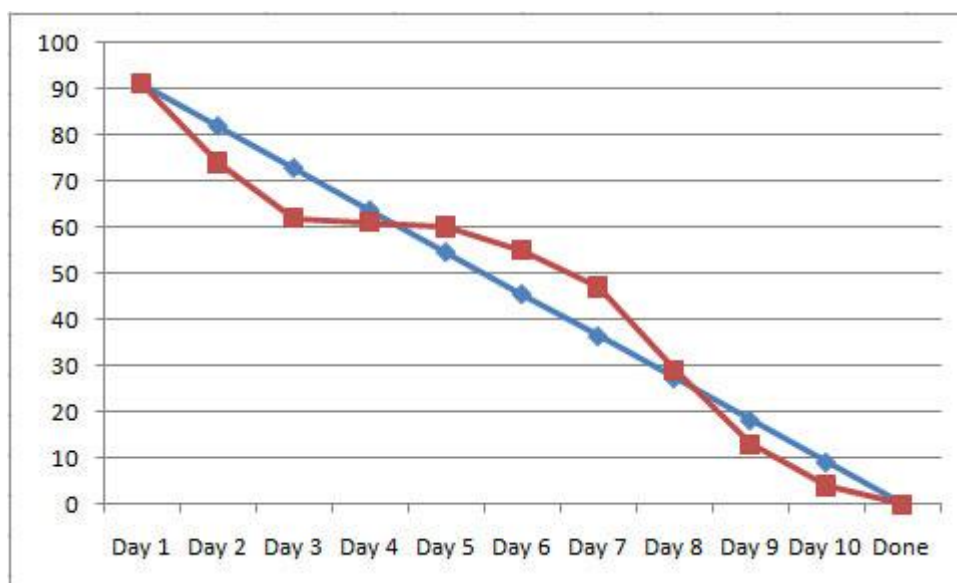


FIGURA 3: Exemplo de um gráfico de burndown. A reta em azul representa a quantidade de horas restantes para cada dia. A linha em vermelho representa o andamento da equipe.

Com a visualização do gráfico é possível ter uma visão palpável do andamento das atividades inseridas no *sprint*. No exemplo da figura 2, as estórias iniciais foram finalizadas antes do tempo diário estimado. O que depois foi revertido com um atraso observado nos dias 5 a 7, até a finalização das tarefas dentro do prazo no dia 11. Caso as estórias fossem classificadas com um peso menor do que o observado, a tendência da linha real é de ficar abaixo da linha base, e estórias com peso maior do que o estimado ou que passaram por algum problema não esperado, tendem a ficarem acima da reta base.

2.4 Perspectiva Isométrica

A perspectiva de objetos é a representação gráfica de objetos tridimensionais que se assemelham a perspectiva de visualização do olho humano. Estas perspectivas podem ser classificadas de diversas maneiras, podendo ser classificadas em perspectivas isométricas, cavaleiras e cônicas (SINHA e UDAI,2008).

A classificação isométrica se baseia na representação de um objeto tridimensional que está localizado no centro dos eixos cartesianos, formando entre si ângulos de 120° (Figura 4) onde cada eixo representa uma dimensão do objeto. Graficamente os objetos são geralmente representados através da inversão dos eixos x e y do plano cartesiano, com um dos vértices inferiores localizado na origem

dos eixos, permitindo assim uma melhora na visualização dos objetos representados (SINHA e UDAI,2008).



FIGURA 4: Representação gráfica das perspectivas isométricas. (a) Perspectiva cônica; (b) Perspectiva cavaleira; (c) Perspectiva isométrica.

Para o desenvolvimento de perspectivas isométricas ou para alocação dos objetos em um plano utilizado por um jogo, por exemplo, é utilizada a malha isométrica (SINHA e UDAI 2008). Essa malha consiste em um conjunto de planos geométricos, geralmente de triângulos equiláteros, dispostos no plano principal com a finalidade de aumentar a proximidade da perspectiva dos objetos.

Para o jogo MyCity foi utilizada a malha isométrica baseada em quadrados girados em 90°, com a finalidade de facilitar a alocação dos objetos utilizados no jogo, todos com base quadrada, para a obtenção de uma maior simetria e organização dos objetos dispostos na tela (PAGELLA, 2011). Esta malha pode ser observada na tela principal do jogo, que limita o espaço útil de construção da cidade, auxiliando na identificação de áreas de possível construção e na otimização de espaço. Esta perspectiva junto com um objeto e seus eixos está representada na Figura 5.

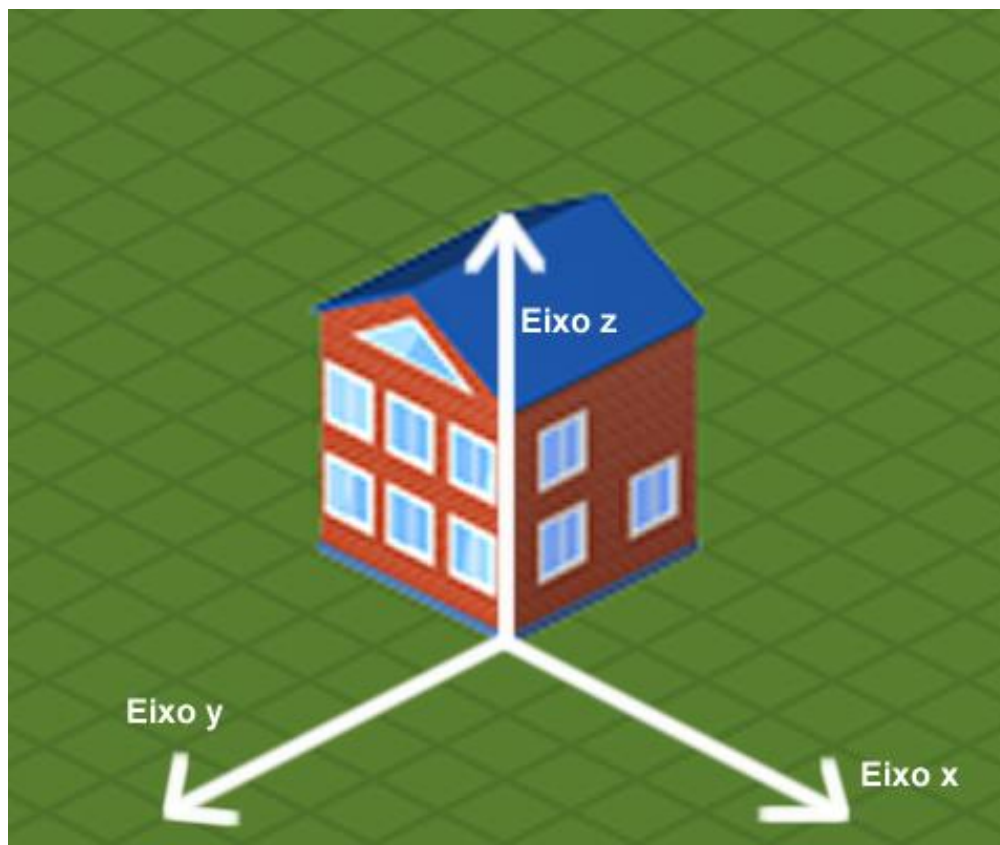


FIGURA 5: Representação da malha isométrica e de um objeto isométrico e seus eixos cartesianos para o jogo MyCity.

2.5 Game Design

Game Design é o ato de decidir como um jogo deve ser (SCHELL, 2008). Consiste na fase de pré-produção de jogos, onde se devem definir detalhes da produção, como o plataformas suportadas, história, cenário onde a história se passa, elementos presentes (Construções, armas, veículos, etc.), personagens e estilo de arte.

Não existem padrões que definam o que devem ou não conter um documento de game design, até mesmo pela grande variação das características que os jogos podem apresentar, o que pode ser fundamental para um projeto, pode ser dispensado em outro.

2.6 Facebook

O desenvolvimento de aplicativos para a rede social Facebook pode ser realizado com várias linguagens de programação web, tendo o apoio de uma grande documentação de sua estrutura e de sua API, e além disso é suportado por ferramentas para depuração e testes dentro do próprio Facebook.

Um usuário interessado em desenvolver aplicativos deve inicialmente registrar-se na área de desenvolvedores do Facebook(<http://developers.facebook.com>). Uma vez registrado, ele tem a liberdade de criar novos aplicativos ou então de poder ingressar na equipe de desenvolvimento de algum aplicativo já criado.

Aplicações podem se integrar de várias formas com o Facebook mas iremos nos ater apenas sobre as aplicações dentro do Facebook. Este tipo de aplicação terá todo o seu conteúdo introduzido dentro de um iframe disponibilizado pela rede social, o qual é chamado de canvas page o qual pode ser populado utilizando qualquer linguagem web como PHP, Python, Java or C# (Facebook Developers).

A estrutura de um aplicativo consiste basicamente de:

- nome
- namespace, nome único para acesso da aplicação
- app id, identificador único da aplicação
- app secret, chave de acesso usado para a autenticação do aplicação
- categoria
- ícone
- canvas url, endereço que terá o conteúdo carregado no iframe

Após criado e configurado, um aplicativo pode ser acessado por um endereço como: <https://apps.facebook.com/nome-da-aplicação>.

2.6.1 Ingresso na aplicação

Para aumentar a interatividade do usuário com a aplicação é interessante recuperar os dados de acesso deste usuário e usá-los juntamente com requisições ao Facebook. Mas para isso a aplicação deve exigir o login do usuário e consequentemente o usuário deve conceder as permissões para acesso de seus dados.

Primeiramente a página que será carregada no iframe deve implementar os passos de autenticação da aplicação. Conforme a linguagem de programação utilizada os métodos e/ou funções utilizadas neste processo podem variar em sua aplicação mas os passos básicos são:

- Configuração da aplicação utilizada: informar app id e app secret necessário para autenticação da aplicação, bem como o endereço da aplicação no Facebook.

- Configuração do escopo das permissões necessárias para a aplicação: Toda aplicação tem acesso às informações básicas do usuário, aquelas que são públicas e que podem ser acessadas por qualquer usuário do Facebook. Ações como postagem em nome do usuário, galerias de fotos e outras ações restritas ao usuário devem ser explicitamente requisitadas suas permissões ao efetuar o login na aplicação.

- Método de login: tal método tem como objetivo recuperar e persistir os dados do usuário conectado na aplicação, uma vez que o usuário tenha permitido a aplicação.

- Método de recuperação do usuário conectado: após o login do usuário ter sido concluído a aplicação deve ser capaz de acessar a identificação do usuário. Como padrão do Facebook, o acesso aos dados de qualquer usuário é realizado através de sua identificação única que possui o nome de *uid*.

Após o usuário ter acessado a aplicação existirá, em sessão, *cookie* ou outra forma de persistência, uma referencia a seu identificador único. Existindo tal referência as requisições à API serão capazes de recuperar e consumir os dados do usuário.

2.7 Considerações Finais

A implementação de um jogo social, do ponto de vista técnico a partir da implementação de visualização isométrica é perfeita, pois sendo para plataforma web o desempenho do aplicativo deve ser otimizado para trabalhar com pouco recurso de hardware. Analisando o projeto como negócio, uma vantagem é percebida pois os jogos do segmento de simuladores urbanos já implementados nas redes sociais não trazem benefícios educacionais, o que faz do uso de indicadores sociais como objetivo do jogador agregar valor, pois proporciona uma experiência mais realista e ética ao ensinar sobre bem-estar social. Para a implementação do projeto a escolha do SCRUM é conveniente para proporcionar entregas parciais e ajudar na interação dos membros da equipe que trabalham em lugares distintos.

3 METODOLOGIA

3.1 Plano de Atividades

O plano de atividades para o desenvolvimento deste projeto foi dividido em seis partes, que seguiram a ordem em que foram executadas durante o andamento do projeto. A divisão pode ser visualizada através do diagrama de WBS que está inserido no apêndice de mesmo nome

A primeira etapa consistiu na elaboração do planejamento inicial, que incluíram a análise do planejamento a ser executado, os riscos e comunicação. Esta parte consistiu em sua maior parte em reuniões com o grupo para discussão de idéias, problemas e soluções para o jogo, além do esboço de protótipos, análise de mercado e perfil de jogadores.

Após a definição da metodologia e a formalização das idéias iniciais, foram levantadas as principais *stories* que seriam desenvolvidas, juntamente com o início da documentação através dos casos de uso e o estudo da API de integração com o Facebook. Após este ponto, foi definida a arquitetura que foi utilizada, incluindo os servidores, linguagens de programação, banco de dados e comunicação cliente/servidor.

O desenvolvimento foi dividido em quatro partes, seguindo os sprints definidos para cada parte. As atividades realizadas em cada sprint estão descritas no apêndice 7.1 descrevendo todas as *stories* executadas.

Nas etapas finais de testes e implantação, foram utilizados testes unitários e por release, além da disponibilização da aplicação em ambiente de produção para a análise de *beta testers* que se prontificaram para a realização de testes e reportar problemas encontrados nas versões lançadas. A última etapa, implantação, foi da análise de impacto do jogo na rede social, acompanhando as críticas realizadas pelos jogadores e a publicação e divulgação de notícias do jogo.

3.2 Responsabilidade

Ainda que todos atuando em todas as áreas do projeto, os integrantes foram incumbidos de gerenciar e atuar a maior parte de suas horas em partes específicas do projeto, como apresentado abaixo:

- Adolfo: desenvolvimento e organização da camada de serviços da aplicação e elaboração dos casos de uso;

- Deividi: elaborar e organizar documentação, elaboração dos diagramas de seqüência e classes e prototipação;
- Felipe: modelagem de dados, manutenção do banco de dados e criação de stored procedures;
- Leandro: desenvolvimento da interface da aplicação, integração com *Facebook*.

3.3 Scrum

3.3.1 Stories

As *stories* com as funcionalidades do sistema foram criadas baseadas nas ações que o jogador terá durante o andamento do jogo e a partir da definição, foram atribuídas os valores de importância e a descrição de cada caso. Com isso foram obtidas divisões das tarefas mais importantes a serem executadas, preferencialmente nos primeiros *sprints* do jogo MyCity, como indicado pela metodologia. Desta maneira evitamos que os desenvolvedores encontrem problemas complexos em fases finais do projeto, que poderiam comprometer os prazos estipulados inicialmente. Estas *stories* foram definidas como demonstrado na tabela 1.

Estória	Novo	Em andamento	Feedback	Fechado
<div>5.001</div> <div>Eu como usuário posso acompanhar meus chamados de redmine</div>	<div>5.010</div> <div>Utilizar serviço para popular a tabela de chamados</div> <div>Renan Paes</div> <div>Normal</div> <div>1.0</div>	<div>5.016</div> <div>Serviço de listagem de chamados</div> <div>André Luis</div> <div>Normal</div> <div>4.0</div>		<div>5.017</div> <div>Utilizar serviço para popular a tabela de chamados</div> <div>Renan Paes</div> <div>Normal</div> <div>1.0</div>
<div>5.002</div> <div>Eu como usuário posso abrir chamados de suporte</div>	<div>5.014</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>	<div>5.018</div> <div>Serviço de abertura de chamados</div> <div>André Luis</div> <div>Normal</div> <div>4.0</div>	<div>5.019</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>	<div>5.023</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>
<div>5.003</div> <div>Eu como administrador posso manter usuários</div>		<div>5.022</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>	<div>5.021</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>	<div>5.025</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>
<div>5.004</div> <div>Eu como usuário posso acessar meus relatórios</div>			<div>5.024</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>	<div>5.026</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>
<div>5.005</div> <div>Eu como usuário posso navegar pelos grupos de aplicativos</div>				<div>5.027</div> <div>Chamar serviço para salvar um novo chamado</div> <div>Adolfo Luis</div> <div>Normal</div> <div>1.0</div>

FIGURA 6: Imagem ilustrativa da ferramenta utilizada

TABELA 1: Definição das *stories* do jogo social MyCity.

<i>Storie</i>	
Nome	Conexão Facebook
Descrição	Como usuário, eu posso me logar ao jogo pelo Facebook
Importância	1
Demonstração	Realizar login no Facebook e acessar o aplicativo, uma vez válido, carregar os seus dados.
<i>Storie</i>	
Nome	Carregar cidade
Descrição	Como usuário, eu posso restaurar minha cidade logo após entrar no aplicativo.
Importância	1
Demonstração	Realizar login no Facebook (storie 1), e logo em seguida, se já tiver criado uma cidade, esta deve ser carregada e pronta para ser jogada.
<i>Storie</i>	
Nome	Criar nova cidade
Descrição	Como usuário, ao acessar o aplicativo pela primeira vez, eu devo ser capaz de criar uma cidade com dados personalizados.
Importância	1
Demonstração	Realizar login no Facebook (storie 1), ao verificar que ainda não possui uma cidade, o aplicativo deve apresentar um formulário onde o usuário pode personalizar sua cidade.
<i>Storie</i>	
Nome	Construir
Descrição	Como usuário, eu devo ser capaz de construir construções.
Importância	1
Demonstração	Escolher construção e clicar em construir. Selecionar a posição desejada no mapa da cidade para construir. Clicar na posição selecionada, e a construção será iniciada. Sistema atualiza os atributos da cidade baseado nos modificadores da construção.

A tabela continua na próxima página.

<i>Storie</i>	
Nome	Alocar operários
Descrição	Como usuário, eu posso alocar operários para construir mais rapidamente alguma construção.
Importância	2
Demonstração	Selecionar uma construção ainda não terminada, adicionar um número x de operários para aumentar a velocidade e clicar em alocar. Verificar que existem x trabalhadores construindo.
<i>Storie</i>	
Nome	Demolir construções
Descrição	Como usuário eu posso demolir qualquer construção que desejar.
Importância	3
Demonstração	Selecionar uma construção, clicar em demolir. Confirmar a demolição. O espaço da construção é desbloqueado.
<i>Storie</i>	
Nome	Convidar amigos
Descrição	Como usuário, eu posso convidar meus amigos do Facebook para jogar comigo.
Importância	2
Demonstração	Acessar o menu integrações, convidar amigos. Selecionar os amigos da lista do Facebook que deseja convidar. Preencher uma mensagem personalizada de convite se desejar. Clicar em convidar amigo. Os amigos selecionados apareceram como pendentes nas notificações de seus amigos.
<i>Storie</i>	
Nome	Alugar operários
Descrição	Como usuário, eu posso deixar a mensagem a todos os meus amigos, pedindo x operários emprestados.
Importância	1
Demonstração	Ao alocar um operário, solicitar aos amigos ajuda com um operário. Essa ação envia uma mensagem a todos os amigos com um botão para confirmar. Ao confirmar, um valor X é debitado da cidade e creditado para o amigo. Imediatamente um operário do amigo é alocado na construção selecionada, ficando ocupado até o termino das obras.

A tabela continua na próxima página.

<i>Storie</i>	
Nome	Empréstimos de dinheiro
Descrição	Como usuário eu posso emprestar algum dinheiro aos meus amigos ou posso pedir dinheiro emprestado.
Importância	4
Demonstração	Acessar o menu empréstimos, dinheiro. Selecionar emprestar dinheiro, a quantidade e o amigo desejado para o empréstimo. Selecionar pedir dinheiro, preencher uma mensagem referente ao pedido e clicar em pedir aos meus amigos.
<i>Storie</i>	
Nome	Presentes
Descrição	Como usuário eu posso enviar construções especiais aos meus amigos e ver minha caixa de correio.
Importância	4
Demonstração	"Acessar o menu presentes, enviar a um amigo. Selecionar um amigo da lista do Facebook que tenha o jogo, selecionar a construção para o presente, preencher uma mensagem e clicar em enviar. Acessar o menu presentes, minha caixa de correio. Verificar os presentes existentes. Responder ao amigo que enviou o presente."
<i>Storie</i>	
Nome	Verificar meus detalhes
Descrição	Como usuário eu posso conferir as características da minha cidade, verificar tamanho da população, quantidade de dinheiro e modificadores de qualidade
Importância	2
Demonstração	A interface principal exibe na parte superior o tamanho da população, quantidade de dinheiro e índice de satisfação da população. Ao clicar no índice de satisfação, é exibido um balão com os detalhes de cada modificador de qualidade.
<i>Storie</i>	
Nome	Atualizar level
Descrição	O sistema deve atualizar o level da cidade.
Importância	
Demonstração	De tempos em tempos, o sistema verificará as mudanças na cidade, como população e status da cidade e calculará a experiência adquirida.

A tabela continua na próxima página.

<i>Storie</i>	
Nome	Atualizar atributos
Descrição	O sistema deve atualizar os atributos da cidade de tempos em tempos.
Importância	1
Demonstração	Conforme as construções e a experiência, além dos status e população, o sistema atualiza os atributos da cidade.
<i>Storie</i>	
Nome	Atualizar população
Descrição	O sistema deve de tempos em tempos atualizar a população da cidade.
Importância	1
Demonstração	Conforme as construções e os atributos da cidade, a população deve ser modificada. Menor diferença entre os atributos, e com relação ao número de população, significam maior crescimento.
Nome	Atualizar operários
Descrição	O sistema deve de tempos em tempos atualizar o número de operários disponíveis na cidade.
Importância	1
Demonstração	Conforme a proporção da população e status da cidade, o número de operários varia. Devem haver o controle sobre os operários que estão trabalhando na cidade, aqueles emprestados e aqueles que a cidade recebeu como empréstimo.
Nome	Configurações
Descrição	Como usuário eu posso ajustar a interface da cidade e quais tipos de notificações serão exibidas.
Importância	5
Demonstração	Através do menu de configurações, poderá aumentar e reduzir o volume do jogo e afastar ou aproximar a visão da cidade. É possível selecionar os tipos de notificações que o sistema pode enviar ao Facebook

3.3.2 Burn down

O gráfico de burn down tem como objetivo ajudar o Scrum Master a verificar o andamento do projeto. Neste gráfico os valores do eixo y representam as horas necessárias para o fechamento de todas as tarefas do *sprint* e o eixo x representa os dias do *sprint*. Para que o gráfico de burn down possa ser utilizado de melhor forma, as atividades para o *sprint* devem estar todas bem definidas, com datas e esforço inicial indicados. Não somente as tarefas devem estar bem definidas como também o lançamento das horas trabalhadas deve ser feito diariamente por todos os

integrantes do projeto. O gráfico de burn down é uma ferramenta poderosa para o gerenciamento das atividades da equipe, porém se dados forem omitidos durante o *sprint* o gráfico gerado não será espelhado no rendimento real da equipe.

Sabendo do resultado esperado de um burn down, podemos observar em nossos *sprints* que existiram falhas nos lançamentos das horas trabalhadas, como comprovado no apêndice “Gráficos de burn down”. Como os integrantes da equipe dispõem de poucas horas de dedicação ao projeto, especialmente durante os dias úteis, e também ao fato de ser o primeiro projeto que os integrantes trabalharam com a metodologia do Scrum, por não possuir o costume de relatar as horas trabalhadas diariamente, e pelo fato do lançamento das horas trabalhadas ocorrer geralmente nos finais de semana, os gráficos não condizem com o trabalho realizado.

No primeiro *sprint* (Apêndice 7.2), onde o grupo de focou nos casos de uso UC01, UC02, UC03 e UC04, foram apenas relatadas as horas trabalhadas, porém, não foram informados os esforços das tarefas, tendo como resultado a inexistência da linha que marca os tempos restantes. No segundo *sprint*, que trata os casos UC05 e UC06, o uso e entendimento do Scrum foi melhorados, porém ainda ocorreram erros no lançamento dos dados. Um fato que podemos destacar foi o surgimento de novas tarefas (linha em verde escuro) no final da primeira semana do *sprint* e nos últimos dias do mesmo. Isso se deve ao fato de termos pouca disponibilidade durante os dias úteis e nos finais de semana podemos reavaliar o que foi feito e erroneamente adicionarmos novas stories em um *sprint* já planejado.

Seguindo com o estudo dos gráficos, no terceiro *sprint* obtivemos um burn down completo. Nesse *sprint*, o foco do grupo foi concluir as pendências anteriores, além de implementar os casos de usos UC07 e UC09. Aprendendo com os erros dos *sprints* anteriores, todas as stories para este *sprint* foram alocadas, não havendo mudanças, o que gerou a reta ideal da queima das horas (linha azul). Como não foi atualizado diariamente com as horas trabalhadas, novamente, em um final de semana, todas as horas restantes foram fechadas. Finalmente um caso que nunca deveria ter sido feito. Em nosso último *sprint*, as stories foram definidas, porém as tarefas não obtiveram lançamento de horas estimadas ou trabalhadas. Por falta de planejamento, este *sprint* ocorreu em um tempo menor do que os outros, o que a princípio deveriam ser de três semanas cada *sprint*, este quarto *sprint* levou pouco

mais de duas semanas, acabando por contemplar apenas a finalização das pendências e a implementação do último caso de uso, UC08.

Em resumo, como uma equipe iniciante ao uso da metodologia Scrum, falhamos em não conseguir gerar gráficos de burn down condizentes com nossos esforços. Isso implica em dizer que nosso comprometimento com a metodologia falhou.

Por outro lado, o processo de *releases* ao termino das etapas contribuiu para que diversas alterações se tornassem evidentes. Um exemplo claro disso é a modelagem do banco de dados, que foi elaborada no primeiro *sprint*, e já para o segundo teve que sofrer diversos ajustes para atender às alterações identificadas. Além disso, a proposta inicial de desenvolver uma camada de webservices em PHP, teve que ser revista devido a performance dos servidores, que implicavam em um tempo de resposta elevado. Com isso, a partir do segundo *sprint*, o grupo optou pelo uso de *stored procedures* que elevaram significativamente a velocidade das operações.

3.4 Plano de Riscos

O plano de riscos foi definido avaliando os fatores ambientais, de mercado e da equipe. Os riscos são classificados em duas partes: riscos de desenvolvimento, detalhados nas tabelas 2 e 3, e riscos de uso, detalhados nas tabelas 4 e 5.

TABELA 2: Plano de riscos para o jogo MyCity. Primeira condição.

Nº	1
Condição	Abandono de um dos integrantes
Data Limite	Final
Consequência	Atraso em todo o restante do projeto, a partir da data de abandono
Ação	Reajustar atividades para os demais integrantes a fim de minimizar a perda
Monitoramento	Responsável do projeto mantém constante contato com os demais integrantes
Probabilidade	Muito baixo
Impacto	Alto
Classificação	4

TABELA 3: Plano de riscos para o jogo MyCity. Segunda condição

Nº	4
Condição	Mudanças na API do Facebook
Data Limite	Todo o ciclo de vida do software
Consequência	Atraso nas atividades que necessitam de integração com a rede social durante o desenvolvimento. Atrasos na manutenção quando em fase de produção.
Ação	Estudo da nova API, verificando quais das funções previamente utilizadas foram afetadas e implantar os novos métodos da API.
Monitoramento	Constante monitoramento das modificações na API por parte do responsável do projeto
Probabilidade	Muito baixo
Impacto	Alto
Classificação	4

TABELA 4: Plano de riscos para o jogo MyCity. Terceira condição.

Nº	2
Condição	Aceitação do mercado
Data Limite	Todo o ciclo de vida do software
Consequência	O servidor da aplicação pode ficar sobrecarregado caso haja mais acesso do que ele suporta
Ação	Migra a aplicação para um servidor de porte maior
Monitoramento	Constante monitoramento de acessos por parte de todos os integrantes da equipe
Probabilidade	Alto
Impacto	Moderado
Classificação	6

TABELA 5: Plano de riscos para o jogo MyCity. Quarta condição.

Nº	3
Condição	Servidor da rede social não disponível
Data Limite	Todo o ciclo de vida do software
Consequência	Impossibilidade dos usuários acessarem, caso o software já esteja funcionando, e impossibilidade de desenvolvimento do software caso esteja na fase de desenvolvimento.
Ação	Caso o software esteja em desenvolvimento o responsável deve reajustar as atividades com os integrantes a fim de minimizar o tempo perdido.
Monitoramento	Imediato aviso, por parte dos desenvolvedores, caso o servidor não esteja disponível.
Probabilidade	Muito baixo
Impacto	Alto
Classificação	4

3.5 Plano de Custos

Para os custos de pessoal, foi realizado um estudo dos casos de uso que utilizando a métrica da estimativa de esforço por pontos de caso de uso levantou que o tempo de desenvolvimento seria de aproximadamente 246 horas. Utilizando a constante de 6 horas de trabalho por dia, estipulamos que o projeto será realizado em no máximo dois meses, e que a equipe responsável por ele ser formada por quatro profissionais com remuneração de 4000 (quatro mil) reais.

Para a infra-estrutura durante o desenvolvimento, necessitaríamos de quatro estações de trabalho. A opção para este item foi de quatro computadores Dell Vostro 260s, com Microsoft Office Home and Business 2010 incluso, totalizando de 6000 reais. Para o desenvolvimento e publicação do jogo precisaríamos de ao menos um servidor dedicado e este teria um custo de 1100 reais de aluguel por mês. Além disso, existe a necessidade de um escritório para o local de trabalho, com mobiliário incluso, com aluguel de 900 reais.

Para os softwares necessários para o desenvolvimento optamos por muitas soluções open-source e gratuitas, podendo assim diminuir os custos com estas ferramentas. Independentemente destas escolhas, precisaríamos obter ao menos duas licenças para a ferramenta Adobe Flash Builder 4.5 Standard Edition, necessária para a codificação do esquema do cliente do jogo (a interface para o usuário)

realizada nas linguagens Flex e Action Script 3.0, e estas duas licenças comerciais custam aproximadamente 498 dólares, ou, em média, 855 reais. Como outras ferramentas utilizadas, mas que não são pagas podemos citar a IDE Eclipse para PHP, utilizada na codificação do esquema de servidor do jogo, Astah Community, utilizada para a modelagem dos diagramas UML, MySQL Workbench- Manipulação do banco de dados e elaboração dos diagramas de DER, Gimp, para a manipulação de imagens e Audacity, utilizado para a manipulação de efeitos sonoros.

A definição da porcentagem sobre o custo atual para atuar como reserva de risco, ou seja, definimos que 5% do que será gasto irá ser colocado num fundo que em caso de algum contratempo ou imprevistos possam ser utilizados a fim de não prejudicar o cronograma e entrega do projeto. Além do fundo de risco estipulamos que cerca de 15% do custo total do projeto será utilizado para custear os gastos variáveis (luz, água, telefone, etc.) no projeto.

Levando em consideração a estimativa de tempo gerada pela análise por pontos de caso de uso, o levantamento da equipe e de toda a infra-estrutura necessária para a execução do projeto em um ambiente real de trabalho bem como todos os gastos variáveis e recursos de fundo de risco, o custo total para o projeto seria de aproximadamente 41.000 reais.

3.6 Materiais

3.6.1 Linguagem de programação PHP

É uma linguagem de scripts executada no servidor, classificada como server-side script. Quando um usuário acessa uma página PHP o script é executado no servidor e o resultado é enviado como resposta da requisição. A escolha da linguagem se deu pela ampla comunidade de desenvolvedores, baixo custo de manutenção e pelo grande dinamismo no desenvolvimento.

3.6.2 Doctrine

O Projeto Doctrine é um ORM (*Object Relational Mapper*) livre para PHP apoiado em uma camada de abstração de dados DBAL (*Database Abstraction Layer*).

3.6.3 AMFPHP

É um projeto de software livre para PHP que visa expor métodos de acesso a dados através do protocolo AMF. Esses métodos são consumidos por aplicações

desenvolvidas em Flash servindo como fonte de dados e ponte de comunicação com a aplicação de servidor.

3.6.4 Adobe Flex

É um framework em Action Script 3 que agrega um conjunto de componentes para o desenvolvimento rápido de aplicações para desktop, web e dispositivos móveis. Para a realização de jogos web, executados dentro do navegador, ainda é a melhor escolha pois tem a maior base instalada de players e um vasto número de frameworks e arquiteturas para jogos.

3.6.5 MySQL

O MySQL é o banco de dados livre mais popular do mundo e conta com boa performance, estabilidade e facilidade. Possui baixo custo de manutenção e todos os recursos necessários para a implementação do projeto.

3.6.6 As3Isolib

É um framework livre feito em Action Script 3 para a implementação de ambientes isométricos, como games por exemplo, voltados para a plataforma Flash.

3.6.7 API Facebook

A API do Facebook é formada por um conjunto de Web Services para o consumo e inclusão de dados de usuários. Para facilitar o acesso a essa API o Facebook disponibiliza um conjunto de classes para diversas linguagens, inclusive PHP e JavaScript, que foram utilizadas no projeto.

3.6.8 Scrum

Para a criação e acompanhamento dos sprints e stories criadas, bem como a geração de gráficos de burn down e Gantt (datalhados nos apêndices Gráficos de burn down e Fluxo de Gantt, respectivamente), foi utilizada a ferramenta livre RedMine.

4 APRESENTAÇÃO DO SOFTWARE

4.1 Game Design

4.1.1 Objetivo

O jogo é um simulador onde o jogador, que atua como prefeito, deve construir e administrar uma cidade evitando que ela entre em falência e mantendo a população feliz através dos índices de qualidade de vida. Atividades que o jogador pode desempenhar foram detalhadas no apêndice Diagrama de Atividades.

4.1.1.1 Características

Em oposição aos jogos do gênero disponíveis nas redes sociais o MyCity traz para o jogador uma experiência mais administrativa e estratégica onde cada movimento causa uma reação positiva ou negativa no objetivo do jogo. Cabe ao prefeito buscar o equilíbrio e decidir o que é melhor para a população. O jogo não exige experiência anterior com outros jogos e nem agilidade ou reflexo, sendo assim, é fácil de jogar e cabe a qualquer faixa etária.

4.1.1.2 Critérios de Classificação

O MyCity é baseado em turnos e critérios como segurança, lazer, sustentabilidade, educação e saúde são avaliados ao final de cada turno, impactando no índice de felicidade da população. Esses critérios possuem pesos diferentes e foram escolhidos e baseados na pirâmide de Maslow, uma hierarquia das necessidades que os homens buscam satisfazer segundo o psicólogo Abraham Maslow. Segundo Maslow, quanto mais próximo da base, mais importante uma característica se torna para o indivíduo.



FIGURA 7: Ilustração da Pirâmide de Maslow mostrando a prioridade das necessidades.

Tendo em vista a atual comoção com assuntos ambientais decidimos incentivar a sustentabilidade como um dos critérios, pois acreditamos que ela também colabora para a satisfação dos indivíduos residentes de uma cidade.

4.1.2 Cenário

O cenário é fixo, sem fases e se passa sempre dentro da mesma cidade. A evolução do jogo acontece por um nível de experiência que aumenta conforme o jogador interage. Apenas uma parte dos itens a serem construídos na cidade está disponível no início e conforme esse nível de experiência evolui mais itens são liberados.

Todos os índices e valores importantes para o jogador saber como está o andamento de seus objetivos estão a mostra por meio de painéis na interface. O jogador pode acessar informações sobre qualquer construção a partir de um clique na mesma. Um painel de ações do jogador está presente para que ele possa construir, demolir, convidar ou visualizar amigos e vizinhos, além de alterar configurações do jogo.

A cada 5 turnos é feito um balanço financeiro na cidade para se debitar os custos de manutenção e creditar os lucros que cada construção prove para o prefeito. Tendo o caixa positivo o dinheiro pode ser usado para novas melhorias na cidade, sendo negativo o jogador pode solicitar um empréstimo para um vizinho.

O vizinho é todo aquele amigo integrante da rede social em que o jogo é executado que foi convidado a jogar pelo jogador.

4.1.3 Construções

A evolução do jogo acontece conforme o prefeito constrói ruas, casas, lojas, escolas. Cada construção tem um impacto direto na cidade e está classificada pelos atributos que possuem em comum. A mecânica do posicionamento das construções está descrita nos casos de uso UC05, UC06 e UC07, e ainda no Diagrama de Estados.

4.1.3.1 Residencial

São construções que geram um determinado número de vagas de moradia, conforme o tamanho, e possuem custo mensal.

4.1.3.2 Comercial

Provê vagas de trabalho e geram lucro mensal proporcional com a sua ocupação. A ocupação depende da relação direta entre a população economicamente ativa e o número de vagas de trabalho.

4.1.3.3 Industrial

Disponibiliza mais vagas de trabalho que o comércio, mas também pode gerar um impacto negativo na sustentabilidade e na saúde da população, dependendo da quantidade de itens inclusos na cidade.

4.1.3.4 Estatais

Não alteram a relação trabalhadores/vagas mas agem nos índices de satisfação da população como saúde, segurança e educação. Construções estatais podem ser escolas, hospitais, delegacias, corpo de bombeiros.

4.1.3.5 Decorativas

Construções de infra-estrutura ou de decoração da cidade. Geram custo mensal e podem somar para os índices de lazer e sustentabilidade.

4.1.4 Estado Inicial

Para garantir que o jogador tivesse condições iniciais de jogo, foram efetuados vários experimentos com diferentes configurações para início de jogo, o resultado selecionado foi o que permitiu que os jogadores tivessem um início com pequeno lucro e pequeno potencial de crescimento populacional, mas que necessitassem obras imediatas para maior crescimento. Com base nesses experimentos adotamos as seguintes configurações iniciais:

- 1000 de dinheiro
- 40 de população
- Barracos
- 2 Casas
- 1 Condomínio
- 3 Vendas
- 1 Loja

Essa configuração garante um crescimento de população para até 48 habitantes e oferece ainda 25 empregos, satisfazendo totalmente essa característica, como descrito no item 2.2.

4.1.5 Passagem de Tempo (Turnos)

A passagem de tempo é marcada por turnos executados a cada 10 segundos de jogo. Ações de verificação e validação são executadas pelo servidor com frequências diferentes de turnos. A cada dois turnos os índices de satisfação e a população são atualizados, enquanto que o balanço financeiro só é executado a cada 10 turnos. Essa diferença acontece para evitar que ocorra uma falência financeira por um descuido do jogador, com um período maior o jogador tem mais tempo para desfazer ações.

4.1.6 Socialização

Sendo um jogo imerso dentro uma rede social algumas ações do jogador podem ser veiculadas para a rede em forma de troféu ou conquista, o que leva outras pessoas a conhecerem o MyCity. Benefícios são disponibilizados para jogadores que convidam outras pessoas para jogar (Caso de uso UC08). Além de aumentar a experiência do jogador, contando com mais vizinhos ele aumenta a chance de receber ajuda e presentes quando precisar.

4.1.7 Referência

A principal referência na idealização do jogo é sem dúvida a série de jogos Sim City, de onde foi baseada a mecânica de construções e classificação das construções. Um segundo título de grande influência é a série Tropicó, esse jogo, diferente do SimCity, tem como foco a satisfação da população e serviu de base aos nossos critérios de satisfação.

Em um segundo momento, iniciamos pesquisas em jogos sociais e dois títulos se destacaram: *Farmville* e *Cityville*, especialmente o último, por atuar no mesmo ambiente abordado.

4.2 Telas do Software

Para acessar a aplicação, é necessário realizar a autorização padrão do Facebook para execução de aplicativos que não são nativos da rede social. Após esta etapa, o aplicativo fica automaticamente disponível para o acesso do usuário.

Ao acessar a cidade, a tela do jogo é carregada com todos os atributos existentes da cidade, considerando uma cidade já existente. Por exemplo, temos a tela inicial apresentada na Figura 6 para uma cidade existente.



FIGURA 8: Tela inicial do jogo MyCity.

A tela da Figura 7 consiste na base do jogo, onde é possível visualizar todas as construções, a barra superior com os principais dados da cidade, tais como dinheiro, população, quantidade de operários e satisfação geral da população.

Ainda na Figura 7 é possível observar a barra de ferramentas inferior, ao lado esquerdo da tela, onde o jogador poderá interagir com o mapa, adicionando ou removendo construções, alocando operários e interagir com os amigos da rede social.

A barra superior contém além dos principais dados da cidade os atributos de satisfação da população. Ao pressionar o botão “Sorriso” a janela com estes atributos é exibida. Nela estão apresentados os dados dos itens de Segurança, Saúde, Educação, Sustentabilidade, Lazer e Estética. A Figura 8 apresenta o detalhamento desta opção.



FIGURA 9: Painel superior do jogo MyCity com os atributos de satisfação da população.

O painel inferior apresenta quatro botões de ação para o jogador (Figura 8). O principal deles, o botão de Construir, identificado como 1, é destacado frente aos demais. Ao pressionar este botão, através de uma animação são apresentadas as possibilidades de construções para o jogador selecionar, que pode optar pelo tipo de construção que deseja realizar, com divisões entre Residências, Comércio, Indústria, Estatais e Decoração. Para voltar ao estado normal o usuário pode pressionar o botão Cancelar para o fechamento do painel expandido. Este painel é apresentado na Figura 9.



FIGURA 10: Painel de ações inferior do jogo MyCity.



FIGURA 11: Painel de ações expandido com a opção de Construir selecionada. 1. Abas de tipos de construções; 2. Botões de avançar ou voltar nas opções de cada tipo.

No painel de ações expandido apresentado na Figura 10 é possível navegar entre os tipos de construções através das abas superiores e entre os modelos de cada tipo, através dos botões de avançar ou voltar.

Para informar o jogador sobre as propriedades de cada construção é apresentada uma janela no momento em que o jogador seleciona uma construção. Essa janela contém além do valor da construção, a sua descrição e as propriedades que irão impactar na cidade como um todo, por exemplo, níveis de satisfação da população, lucro ou gastos. Construções como residências apresentam um custo de manutenção, mas aumentam a quantidade de vagas disponíveis na cidade. Uma indústria proporciona uma alta taxa de lucro e empregabilidade, porém diminui na

sustentabilidade da cidade. Neste ponto cabe ao jogador identificar a melhor opção para o desenvolvimento da cidade. A Figura 11 apresenta um exemplo desta janela de detalhes da construção e seus respectivos atributos.

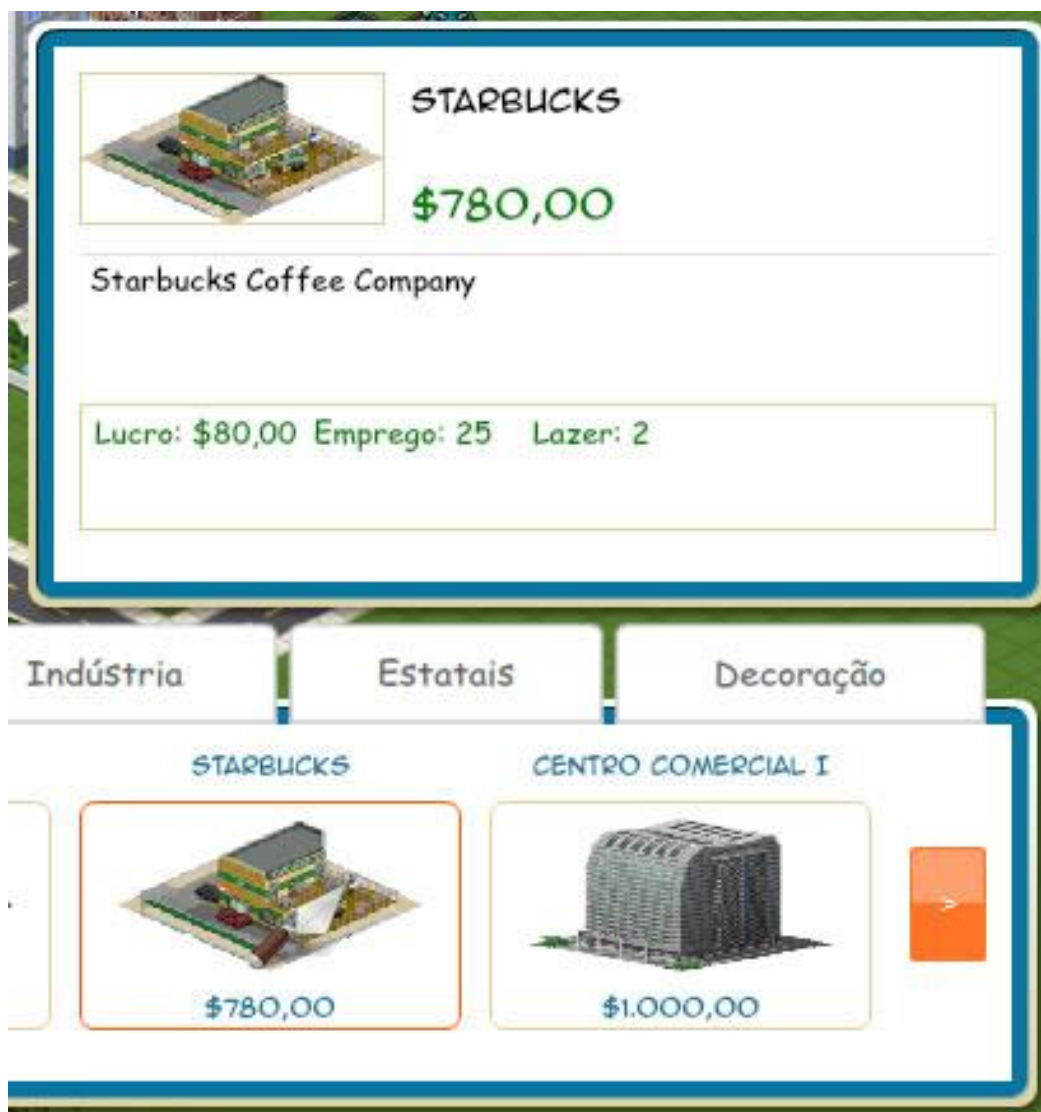


FIGURA 12: Janela de atributos de uma construção, ativada ao selecionar uma construção específica.

Após selecionar uma construção é ativada no mapa a etapa de seleção de terreno para construir o item selecionado. Ao passar o cursor do mouse sobre o mapa é indicado ao jogador se a construção é permitida no ponto escolhido. Caso positivo, ela ficará com a cor verde e, caso a construção não seja permitida, pois irá sobrepor outra construção, ela ficará na cor vermelha. Ambos os exemplos estão demonstrados na Figura 12.



FIGURA 13: Visualização de permissão para construção. A – Construção não permitida, pois está sobreposta com a rua; B – Construção permitida.

Após pressionar o botão direito do mouse em um espaço permitido para a construção, o item escolhido é automaticamente adicionado no mapa do jogo e sua etapa de construção é iniciada e algumas ações ocorrem neste momento. As ações consistem no débito do valor da construção do dinheiro total da cidade e da alocação de um operário para realizar a construção. A partir deste momento é exibido um balão sobre a construção que indica a quantidade de ciclos que faltam para que esta construção seja finalizada. É possível que o jogador aumente a quantidade de operários alocados para esta construção, fazendo com que ela finalize com uma menor quantidade de ciclos necessários, que pode ser visualizada na Figura 13.



FIGURA 14: Andamento de uma construção. A – Barra com a quantidade de ciclos realizados em relação ao total para completar a construção; B – Opção para adicionar novos operários na construção.

Após o término da construção a opção de adicionar operário não é mais disponibilizada e, no lugar desta janela, uma nova janela de informações é apresentada ao clicar na construção. Com esta janela é possível observar, por exemplo, a quantidade atual de lucro, emprego e lazer que esta construção está contribuindo para a cidade, como apresentado na Figura 14.



FIGURA 15: Janela de informações de uma construção finalizada.

Após construir ruas, residências ou comércios, por exemplo, o jogador pode remover os itens que deseja, para substituí-los ou reorganizar sua cidade utilizando a opção de Demolir (Figura 9 - Item 2). Ao selecionar a opção o mapa entra no modo de demolição e o item que for selecionado será excluído do mapa quando o jogador

pressionar o botão esquerdo do mouse. O modo Demolir altera o cursor do mouse com o mesmo ícone do botão Demolir, e o usuário pode identificar a construção que será removida através da borda em vermelho que identifica a construção selecionada. A Figura 15 representa esta ação para demolir em duas etapas, anterior e posterior à demolição.

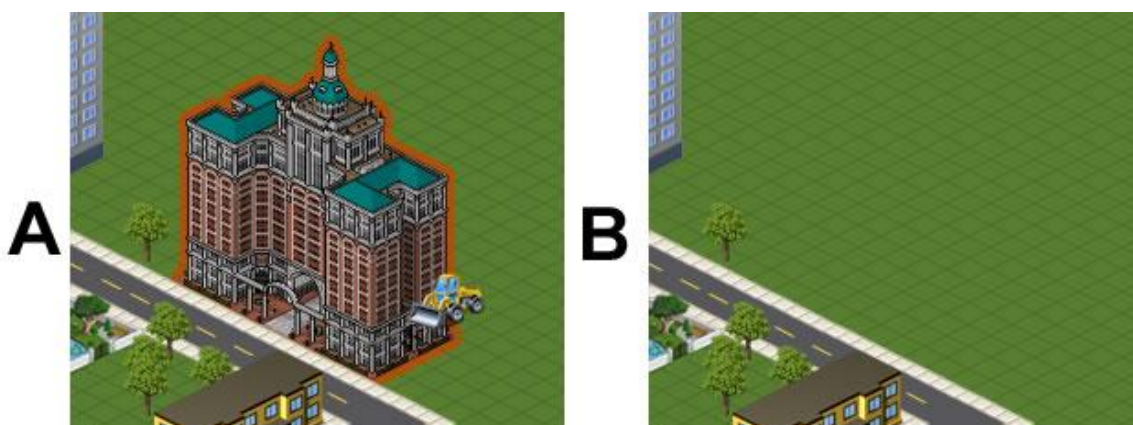


FIGURA 16: Modo de demolição de construções. A – Ao passar o cursor sobre uma construção que será removida; B – Após pressionar o botão do mouse para demolir.

As ruas são um dos principais itens da cidade, que irão interligar as construções da cidade. A interligação de cruzamentos e quadras é realizada automaticamente pelo jogo. O exemplo ilustrado na Figura 16 apresenta este tipo de ligação automática.

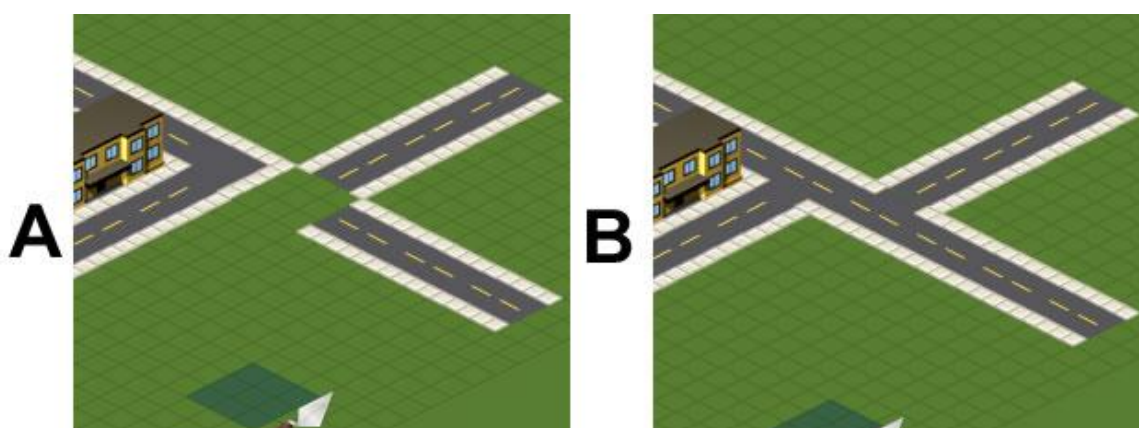


FIGURA 17: Interligação de ruas. A – Ruas isoladas; B – Ruas conectadas automaticamente.

O botão de Amigos do painel de ações (Figura 9 – Item 3) permite que o jogador visualize os amigos do Facebook que estão participando do MyCity e permite que ele chame amigos que ainda não participam do jogo. Ao selecionar este

botão a lista com os amigos é carregada do Facebook (Figura 17), com a imagem do perfil de cada jogador. Para os campos que não possuem amigos, a opção de Convidar Amigo fica disponível para o envio de convites. A seleção de amigos para o envio de convites é uma janela da própria rede social que é acionada quando o botão é pressionado (Figura 18).

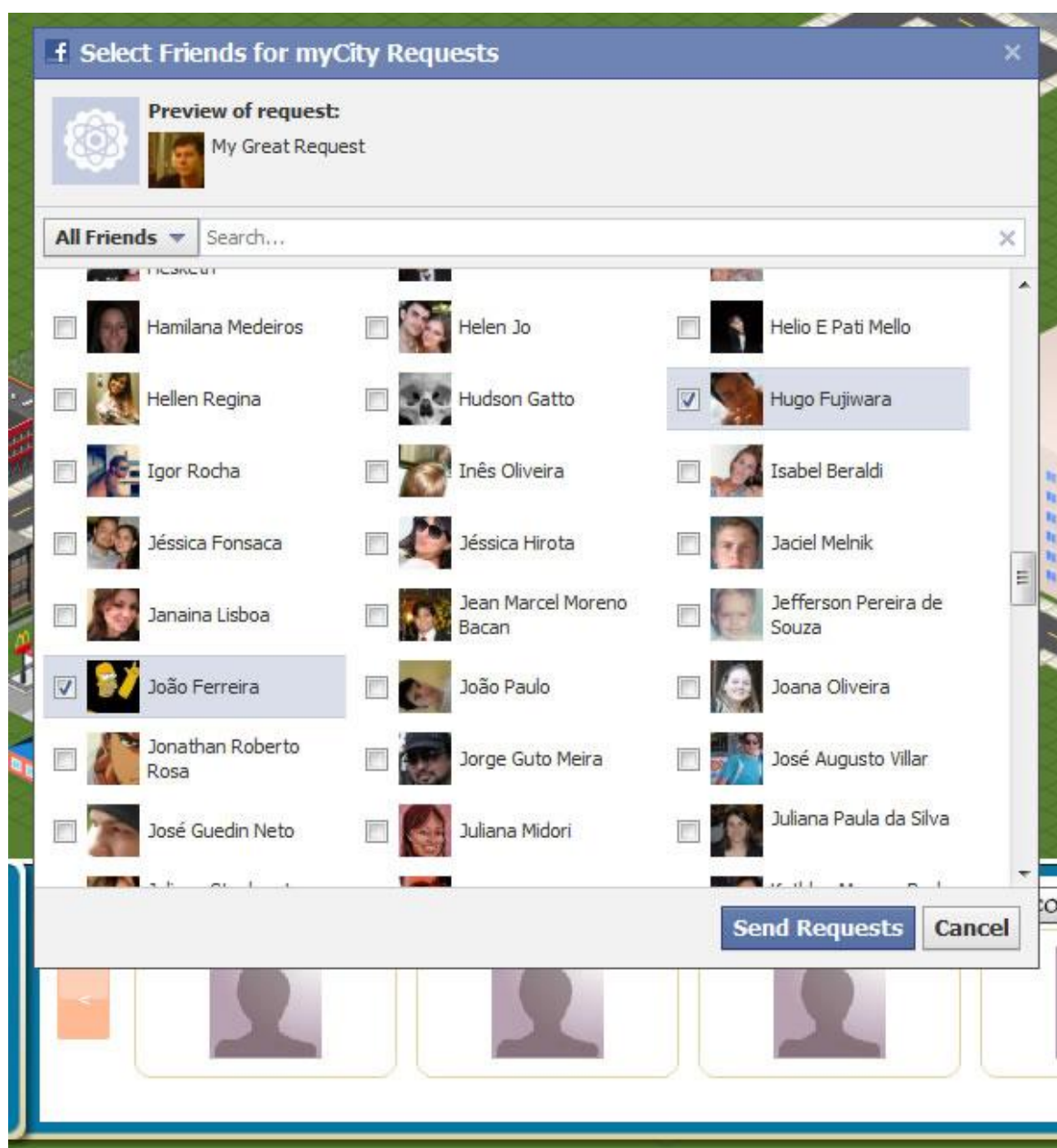


FIGURA 18: Pannel de amigos e convites do Facebook.

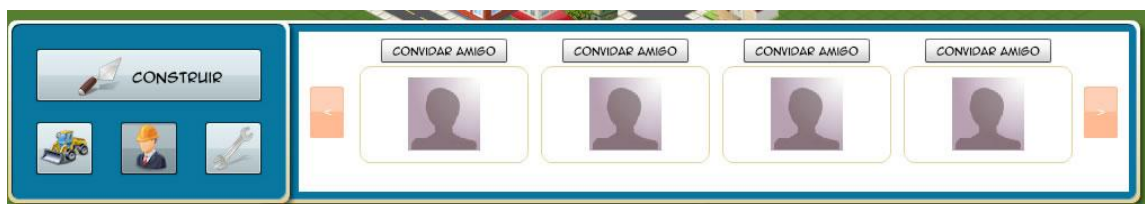


FIGURA 19: Painel de convites do Facebook para adição de amigos no jogo MyCity.

Os convites enviados para outros usuários do Facebook ficam registrados em uma tabela na base de dados e geram créditos para o jogador que indicou assim que um convidado inicia uma nova cidade no jogo.

O último botão do Painel de ações (Figura 9 – Item 4) apresenta as opções do jogo. Estas opções estão apresentadas na figura 19 e consistem no modo de tela cheia, que esconde todos os itens do navegador e do sistema operacional, a opção de Música do jogo e a opção de Sons, que ativa os sons característicos ao executar determinadas ações durante o jogo, como construir ou demolir.



FIGURA 20: Painel de Configurações do jogo MyCity.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho permitiu a criação do jogo *MyCity*, voltado para a utilização em redes sociais, baseada em metodologias ágeis de desenvolvimento. Para isso, a concepção do jogo foi realizada através do mapeamento do mercado de redes sociais e seu potencial, aliado às pesquisas de design, regras do jogo e tecnologias que se adaptam a esse modelo de desenvolvimento.

A metodologia ágil *Scrum* foi de extrema importância durante o desenvolvimento do trabalho, pois proporcionou uma otimização dos recursos disponíveis, como tempo, conhecimentos técnicos e divisão de tarefas. Para isso, todas as etapas de desenvolvimento foram divididas em histórias e classificadas por importância relativa, facilitando na criação dos *sprints* de execução do projeto. Devido a falta de experiência dos integrantes do grupo em reportar suas atividades, os resultados gráficos obtidos com a metodologia não foram satisfatórios para a análise das tarefas realizadas, pois apresentavam valores que não condizentes com a realidade. Dois outros fatores importantes para a falha no processo são a indisponibilidade em se dedicar ao projeto todos os dias, devido a aulas ou mesmo trabalho, o que prejudicava as estimativas e acompanhamento, e ainda a ausência de reuniões diárias de scrum para acompanhamento das atividades.

As linguagens de programação utilizadas foram o PHP, Flex e Action Script, aliados a frameworks que permitiram a criação de um ambiente gráfico e estrutural que possibilitam o jogador de executar o jogo com um padrão de integração e layout similar aos existentes no mercado.

5.1 Melhorias

Num primeiro momento, as melhorias do jogo devem focar em atender jogadores sem experiência, através de tutoriais para os mesmos. Na etapa seguinte o jogo deve se dar na divulgação do mesmo, utilizando para isso fatores de integração com Facebook, como sistema de *rank*, envio de presentes e criação de *fan page* para os jogadores.

Uma melhoria de grande importância para a continuidade do projeto (Capítulo 5.2), trata de ferramentas gerenciais, capazes de oferecer relatórios e mecanismos para manutenção dos itens do jogo.

5.2 Continuidade

O trabalho de desenvolver um jogo como esse é uma atividade que só termina quando o próprio jogo deixa de existir. São necessárias ações contínuas para manter o interesse dos jogadores. Dentre essas ações, três se mostram principais:

1. Adição de novas construções;
2. Adição de novas façanhas;
3. Balanceamento contínuo de quaisquer itens do jogo, garantindo que não existam vantagens injustas entre diferentes itens.

REFERÊNCIAS BIBLIOGRÁFICAS

CAMPI, M. (2011). Facebook passa Orkut no Brasil, diz comScore. Disponível em Info Exame: <http://info.abril.com.br/noticias/internet/Facebook-passa-orkut-no-brasil-diz-comscore-17012012-49.shl>. Acesso em 17 de Janeiro de 2011

DE PAULA, R. (2010). 10 Estatísticas incríveis de games sociais no facebook. Disponível em: <http://www.midiassociais.net/2010/10/10-estatisticas-incriveis-de-games-sociais-no-facebook>. Acessado em 03 de Fevereiro de 2012.

CAPORUSSO, D.; MATIAS, L. F (2008). Áreas verdes urbanas: avaliação e proposta conceitual. Disponível em: <http://www.rc.unesp.br/igce/simpgeo/71-87danubia.pdf>. Acesso em 21 de Janeiro de 2012.

DYBA, T.; T. DINGSOYR (2008). "Empirical studies of agile software development: A systematic review." Information and Software Technology, 50, (9-10): 833-859.

FIELDS, T.; B. COTTON (2011). Social Game Design: Monetization Methods and Mechanics, Elsevier Science.

FIELDS, T.; B. COTTON (2012). Chapter Six - What Is a Social Network? Social Game Design. Boston, Morgan Kaufmann: 87-121.

GUZZO, P. (2011) PROGRAMA PRÓ-CIÊNCIAS: ARBORIZAÇÃO URBANA. Disponível em: <http://educar.sc.usp.br/biologia/prociencias/arboriz.html>. Acesso em 21 de Janeiro de 2012

JACKSON, M. O.; A. WATTS (2010). "Social games: Matching and the play of finitely repeated games." Games and Economic Behavior 70(1): 170-191.

LEMOS, L. (2011). Cityville ultrapassa os 100 milhões de jogadores por mês. MyGames e-Zine. Disponível em: <http://ez.mygames.pt/pc/estrategia/cityville/noticia/cityville-ultrapassa-os-100-milhoes-de-jogadores-por-mes-13800/>. Acesso em 20 de Fevereiro de 2012.

MANIFESTO, A. (2001). Manifesto for Agile Software Development. Disponível em: <http://agilemanifesto.org/>. Acesso em 15 de Novembro de 2011.

OCDE (2011) Organisation for european economic co-operation. Disponível em: http://www.oecd.org/document/48/0,2340,en_2649_201185_1876912_1_1_1_1,00.html. Acesso em 21 de Janeiro de 2012.

PAGELLA, M. A. (2011). Making Isometric Social Real-Time Games with Html5, Css3, and JavaScript, O'Reilly Media.

PETC (2011). Primary education, teachers (per capita) by country. World Development Indicators database. Disponível em: http://www.nationmaster.com/red/graph/edu_pri_edu_tea_percap-education-primary-teachers-per-capita&int=-1&id=OECD. Acesso em 21 de Janeiro de 2012.

PMC (2011) Prefeitura Municipal de Curitiba. Guia geográfico: Parques de Curitiba. Disponível em: <http://www.curitiba-parana.net/parques.htm>. Acesso em 21 de Janeiro de 2012.

PNAD (2008). Pesquisa Nacional por Amostra de Domicílios Disponível em: http://www.ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=1455&id_pagina. Acesso em 21 de Janeiro de 2012.

PPCC (2011). Police per capita by country List of countries by size of police forces. Disponível em: <http://www.nationmaster.com/graph/crime-police-per-capita>. Acesso em 21 de Janeiro de 2012

SETC (2011). Secondary education, teachers (per capita) by country, World Development Indicators database. Disponível em: http://www.nationmaster.com/red/graph/edu_sec_edu_tea_percap-education-secondary-teachers-per-capita&int=-1&id=OECD. Acesso em 21 de Janeiro de 2012.

SCHELL, J. (2008). The Art of Game Design: A book of lenses, Burlington, Ed. Morgan Kaufmann.

SHIN, D.-H.; Y.-J. SHIN (2011). "Why do people play social network games?" Computers in Human Behavior 27(2): 852-861.

SINGH, M. (2008) Angry Mumbai Wants Answers, Changes. Disponível em Time: <http://www.time.com/time/world/article/0,8599,1862893,00.html>. Acesso em 20 de Janeiro de 2012

SINHA, S. U. A. D. U.; S. UDAI (2008). Computer Graphics, McGraw-Hill Education (India) Pvt Ltd.

SMTNS (2011). Shopping Malls: Total number by state. National Research Bureau, 2005. Disponível em: http://www.StateMaster.com/graph/lif_num_of_sho_mal-lifestyle-shopping-malls-total-number. Acesso em 21 de Janeiro de 2012

TEIXEIRA, R. (2008). Disponível em: http://www.senado.gov.br/portaldoservidor/jornal/jornal90/comportamento_lazer.aspx. Acesso em 21 de Janeiro de 2012

VLAANDEREN, K., S. JANSES; (2011). "The agile requirements refinery: Applying SCRUM principles to software product management." Information and Software Technology53 (1): 58-70.

VENCESLAU, P. (2010). Saúde e segurança são prioridades para eleitor. Disponível em Grupo Ibope: [http://www.ibope.com.br/calandraWeb/servlet/CalandraRedirect?temp=5&proj=Portal IBOPE&pub=T&db=caldb&comp=Elei%E7%F5es+e+Administra%E7%E3o+P%FAbli ca&docid=E66F2BA21FB8B2D883257745004F7FC0](http://www.ibope.com.br/calandraWeb/servlet/CalandraRedirect?temp=5&proj=PortalIBOPE&pub=T&db=caldb&comp=Elei%E7%F5es+e+Administra%E7%E3o+P%FAbli ca&docid=E66F2BA21FB8B2D883257745004F7FC0). Acesso em 20 de Janeiro de 2012.

WHO (2009). Number of physicians. Disponível em WHO Indicator and Measurement Registry: http://apps.who.int/gho/indicatorregistry/App_Main/view_indicator.aspx?iid=104. Acesso em 20 de Janeiro de 2012.

APÊNDICE – DOCUMENTAÇÃO DO SOFTWARE

Apêndice 1 - Gráficos de burn down.

Apêndice 1.1 Burn Down – Sprint 1

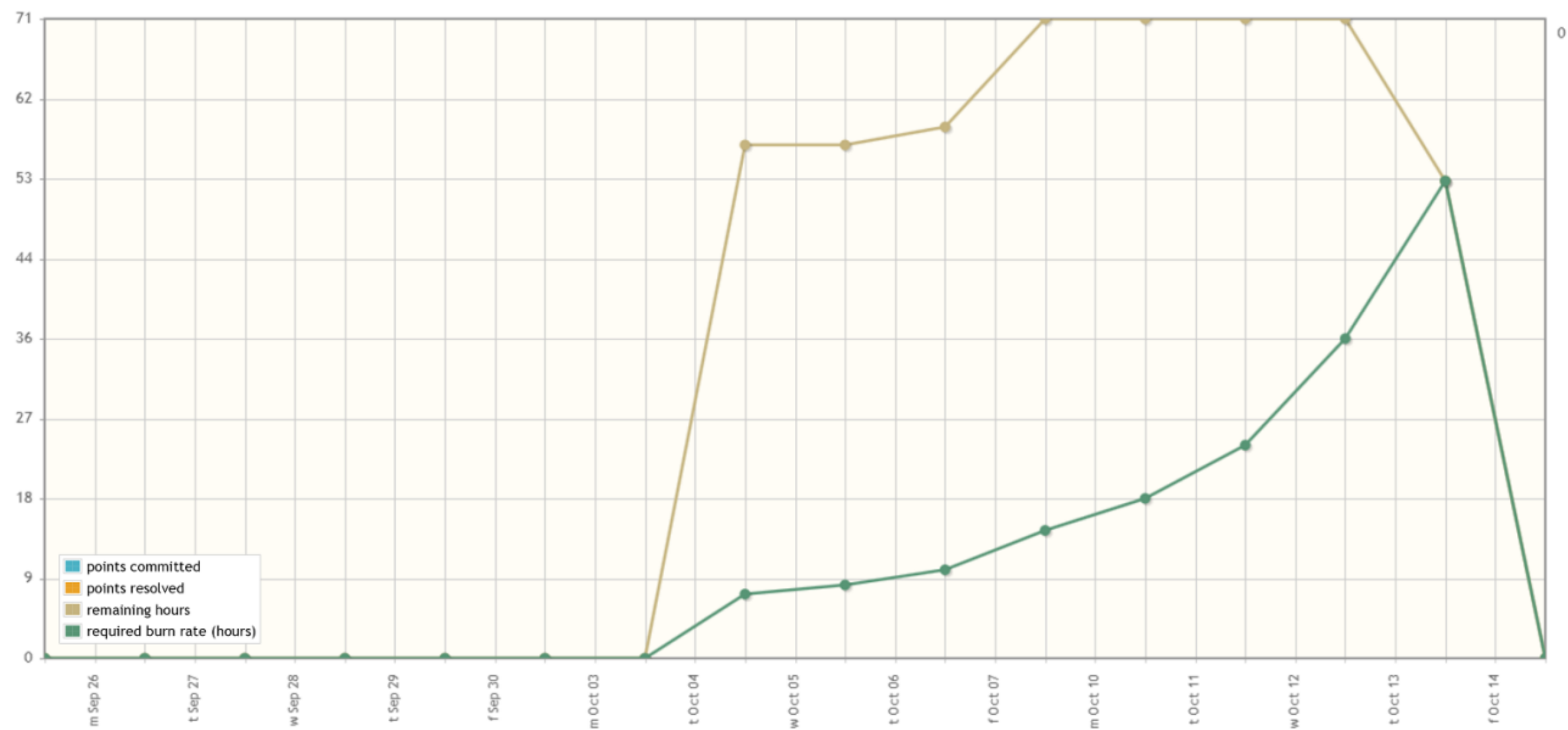


FIGURA 21: Gráfico de Burn Down para o Sprint 1

Apêndice 1.2 Burn Down – Sprint 2

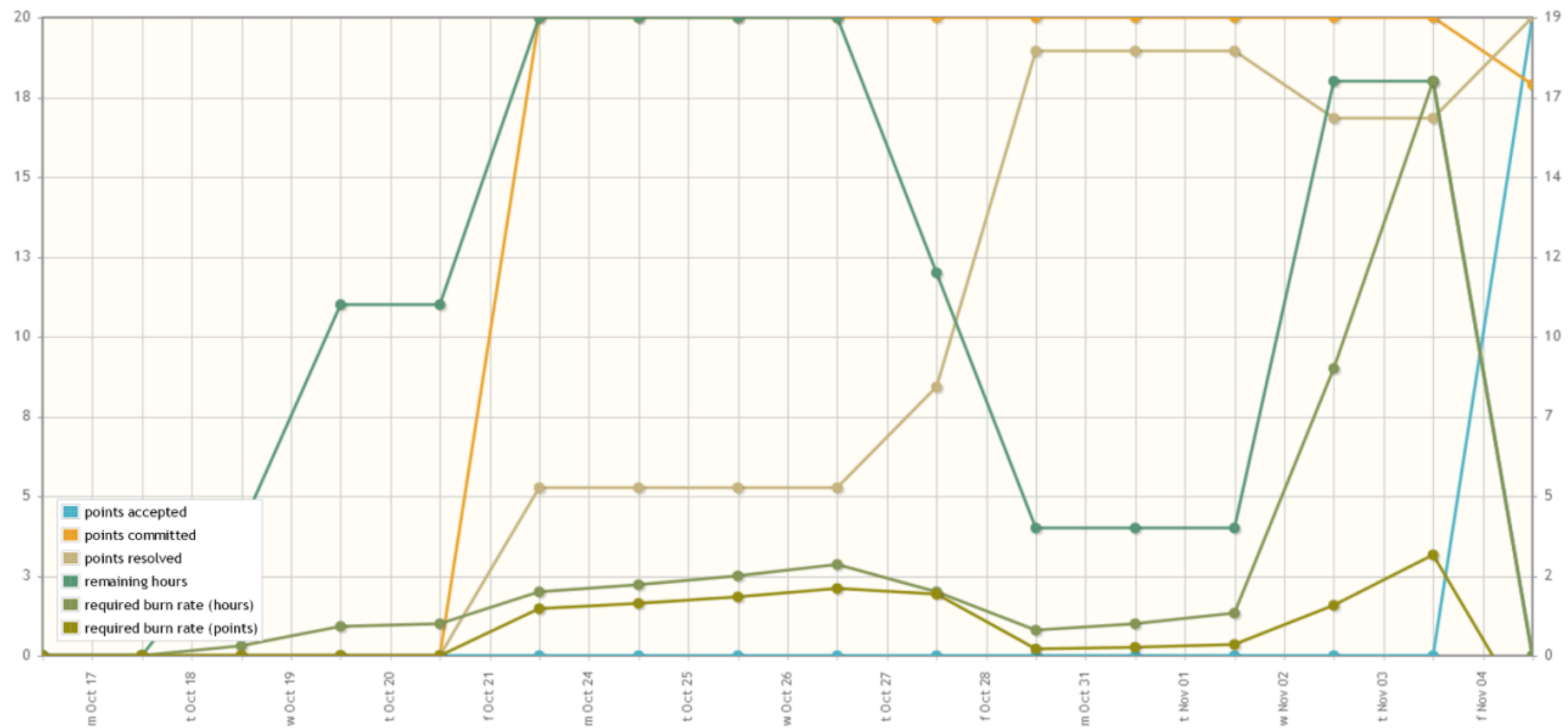


FIGURA 22: Gráfico de Burn Down para o Sprint 2

Apêndice 1.3 Burn Down – Sprint 3

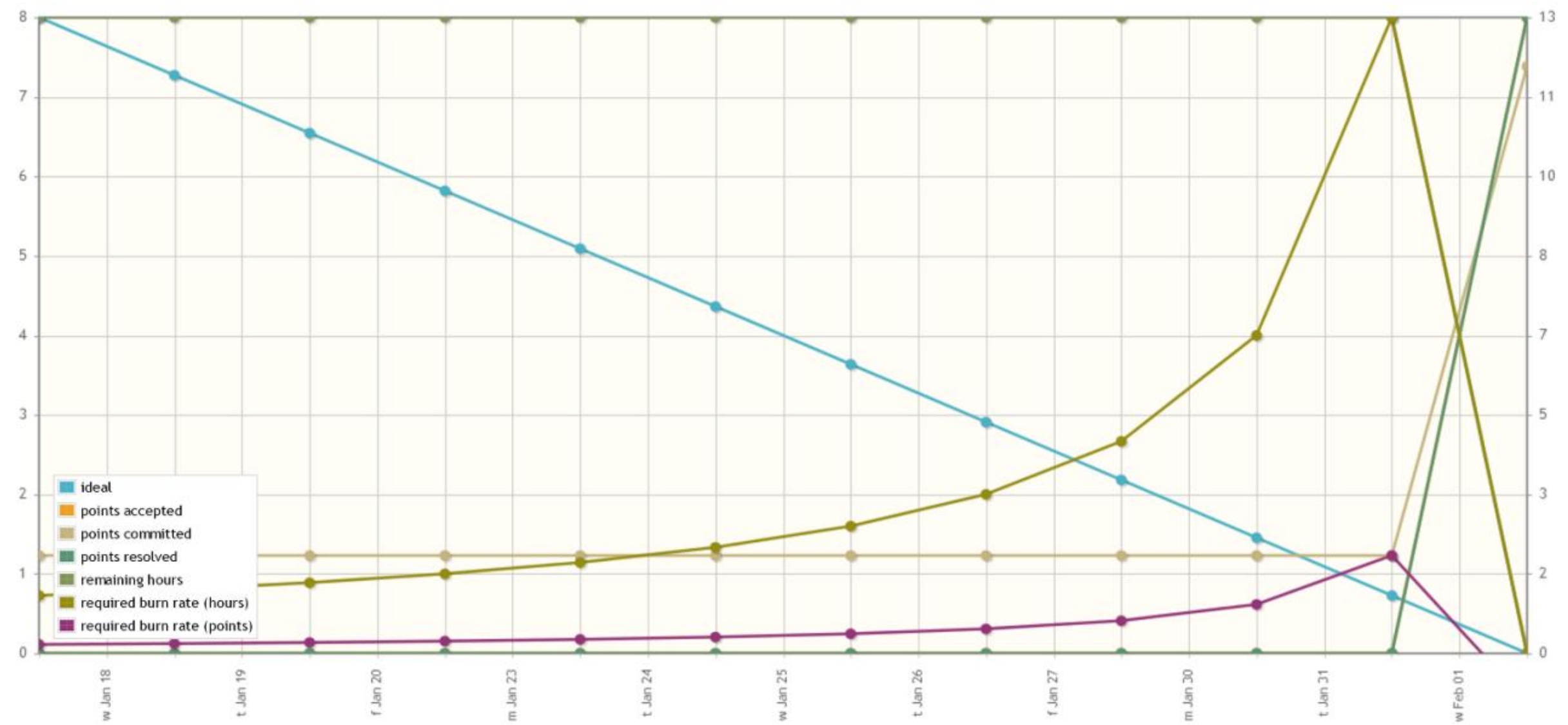


FIGURA 23: Gráfico de Burn Down para o Sprint 3

Apêndice 1.4 Burn Down – Sprint 4

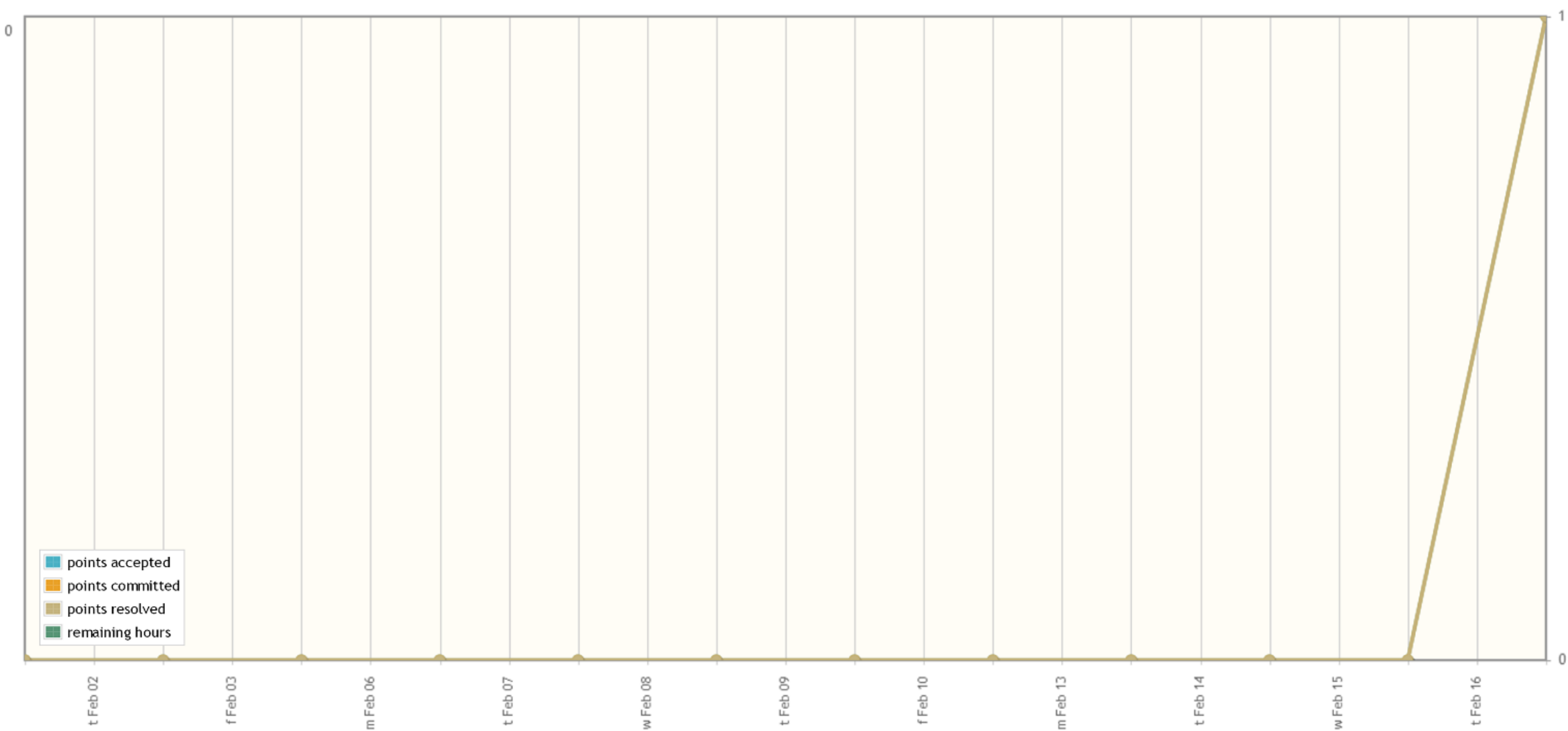


FIGURA 24: Gráfico de Burn Down para o Sprint 4

Apêndice 2 - Diagrama de WBS

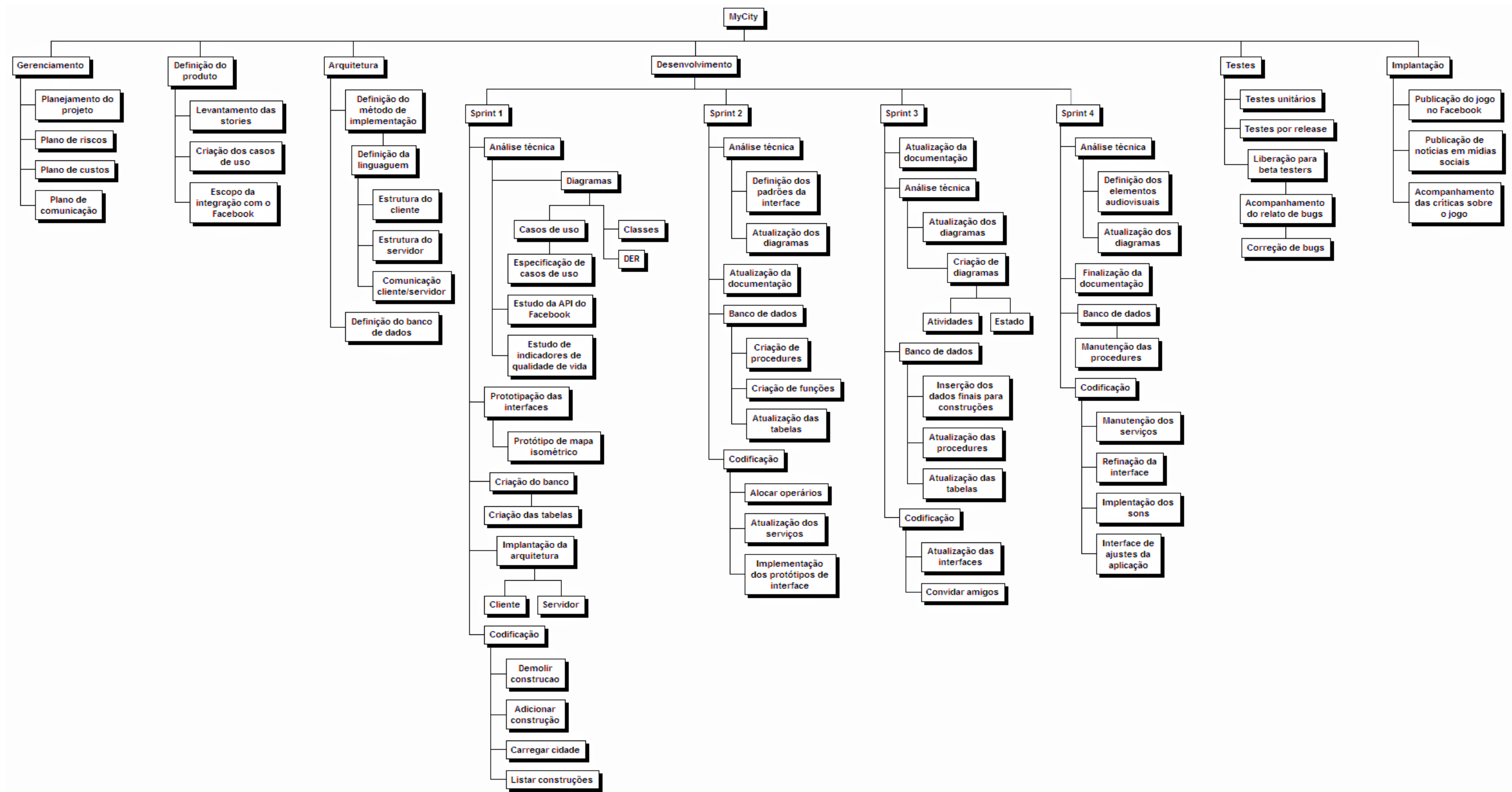


FIGURA 25: Diagrama de WBS para o jogo MyCity.

Apêndice 3 - Fluxo de Gantt

Apêndice 3.1 Fluxo de Gantt para os Quatro Sprints do Projeto

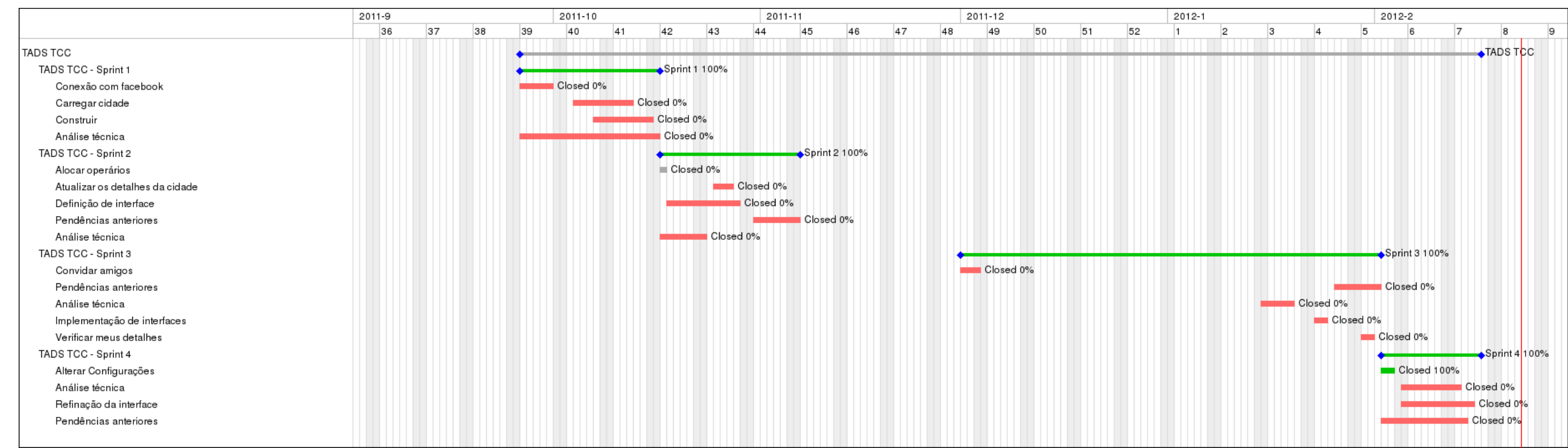


FIGURA 26: Gráfico de Gantt para os quatro sprints do jogo MyCity.

Apêndice 3.2 Fluxo de Gantt para o Sprint 1

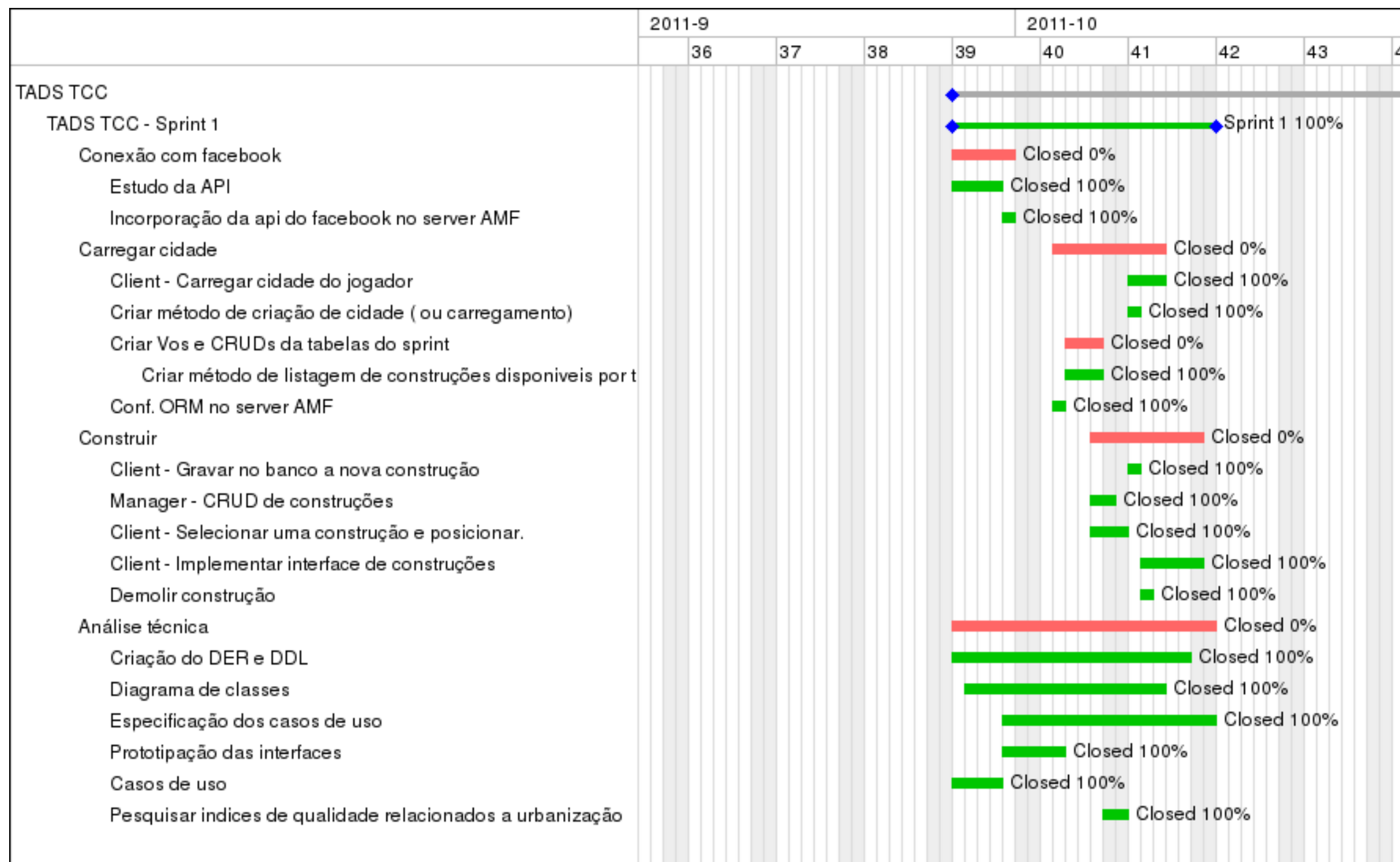


FIGURA 27: Gráfico de Gantt para o Sprint 1.

Apêndice 3.3 Fluxo de Gantt para o Sprint 2

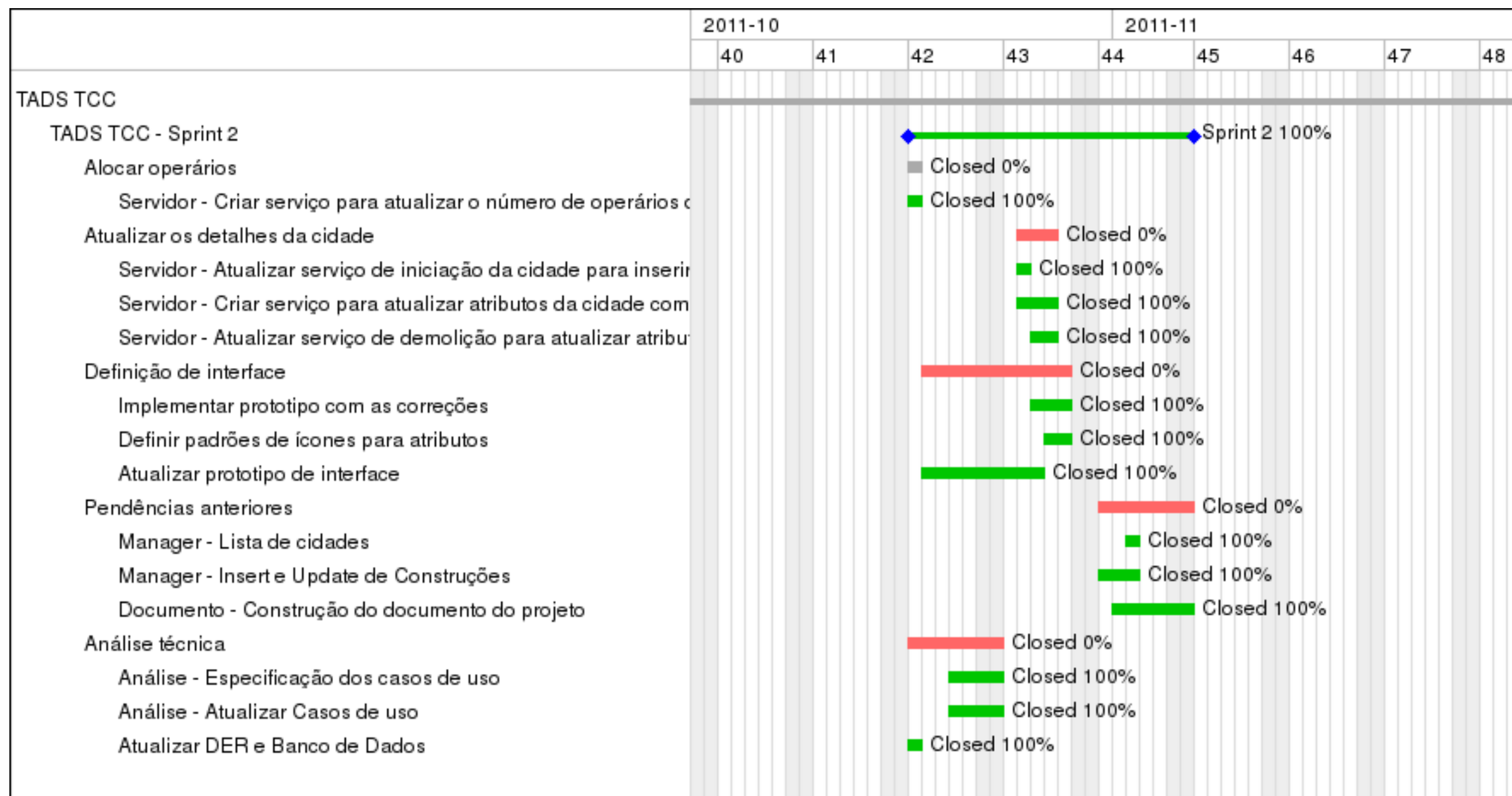


FIGURA 28: Gráfico de Gantt para o Sprint 2.

Apêndice 3.4 Fluxo de Gantt para o Sprint 3

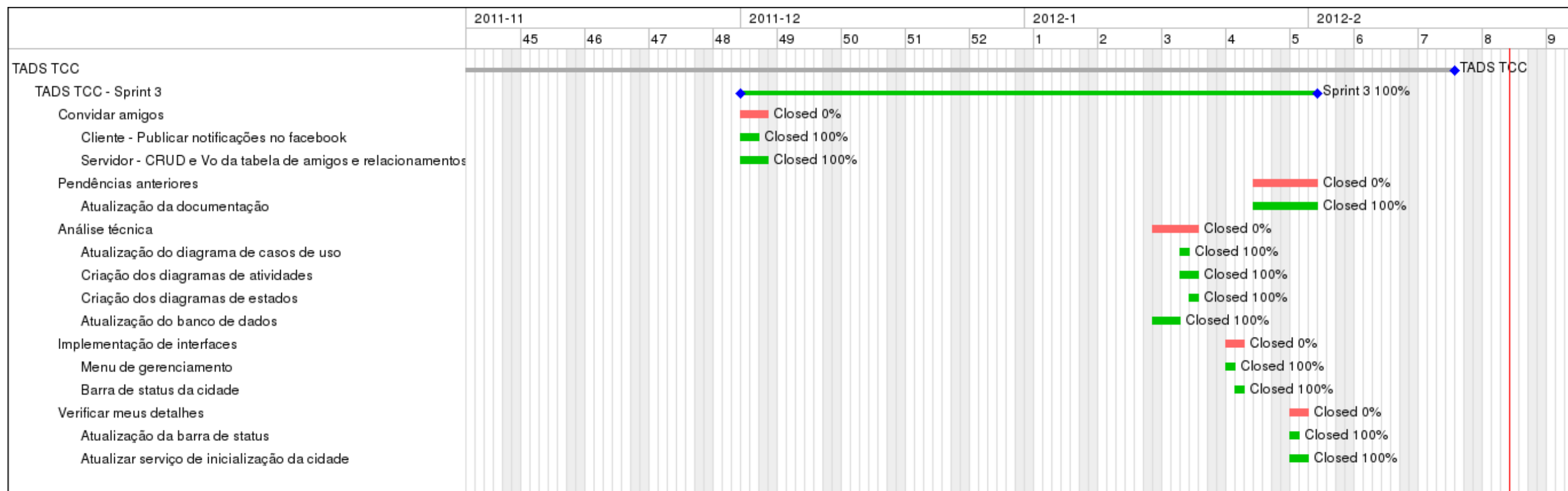


FIGURA 29: Gráfico de Gantt para o Sprint 3.

Apêndice 3.5 Fluxo de Gantt para o Sprint 4

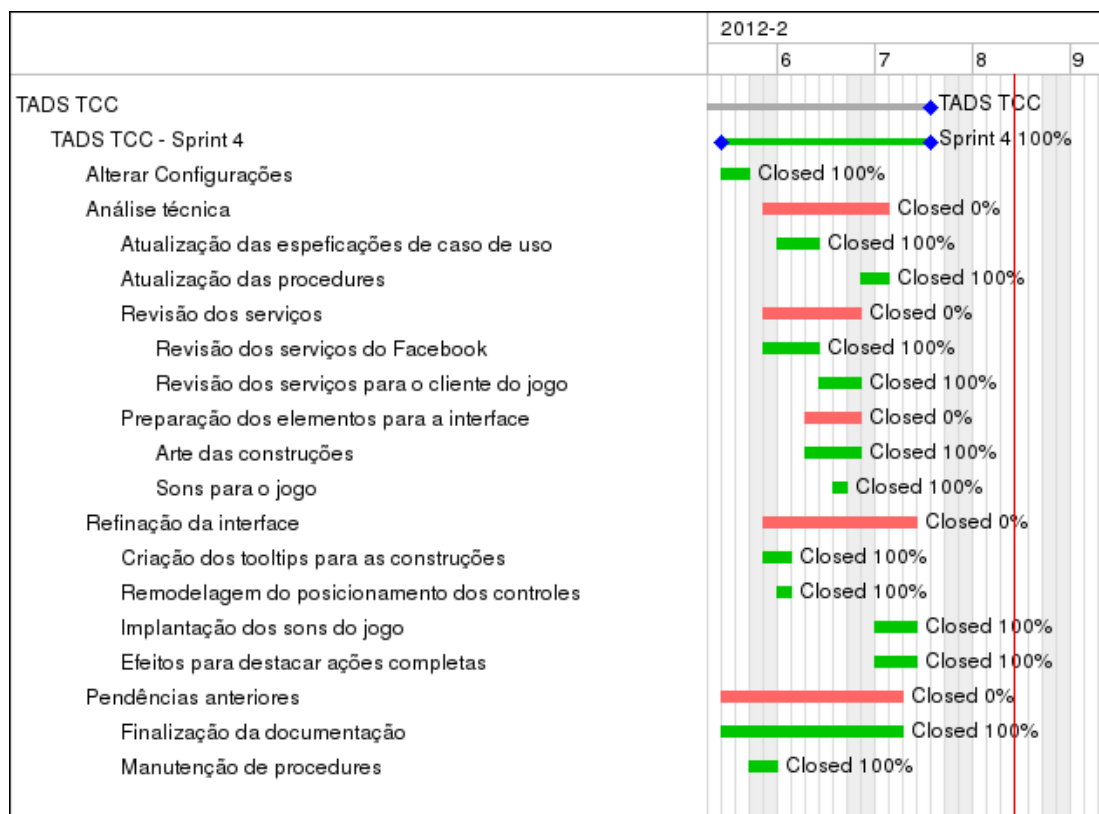


FIGURA 30: Gráfico de Gantt para o Sprint 4.

Apêndice 3.6 Fluxo de Gantt Detalhadopara os Quatro Sprints

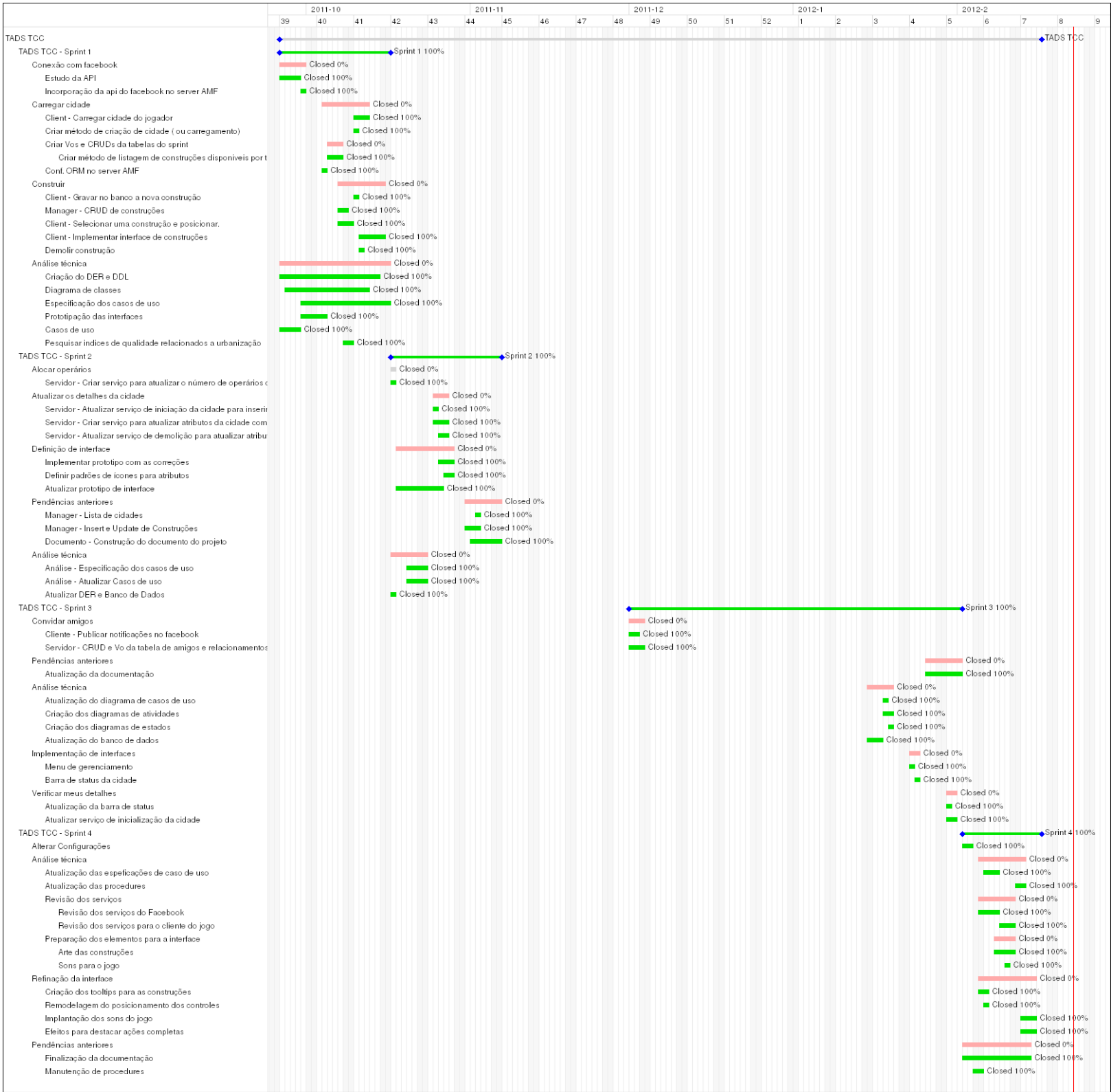


FIGURA 31: Gráfico de Gantt detalhado para os quatro Sprints realizados.

Apêndice 4 - Diagrama de Classes

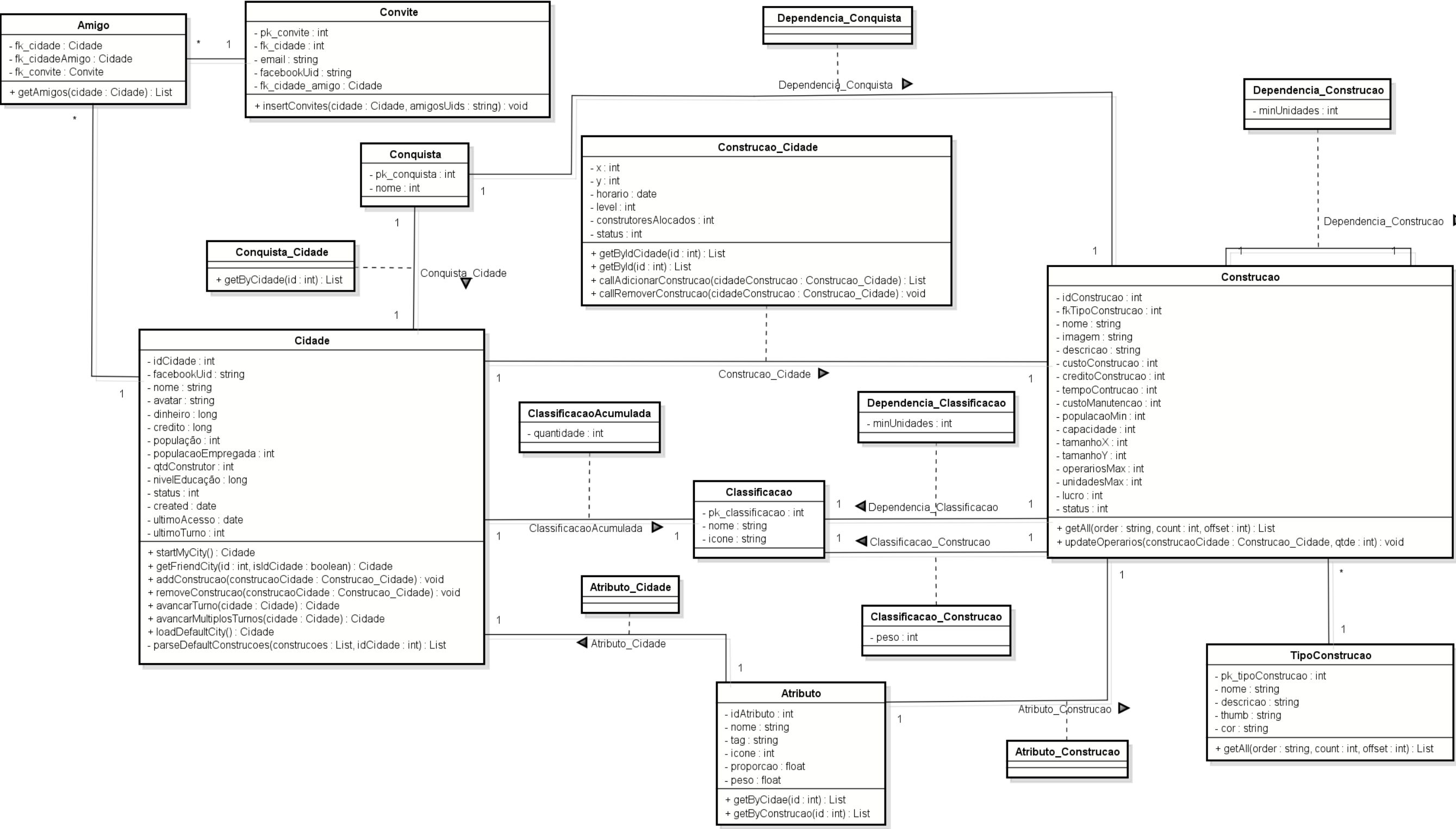


FIGURA 32: Diagrama de Classes.

Apêndice 5 - Diagrama de Casos de Usos

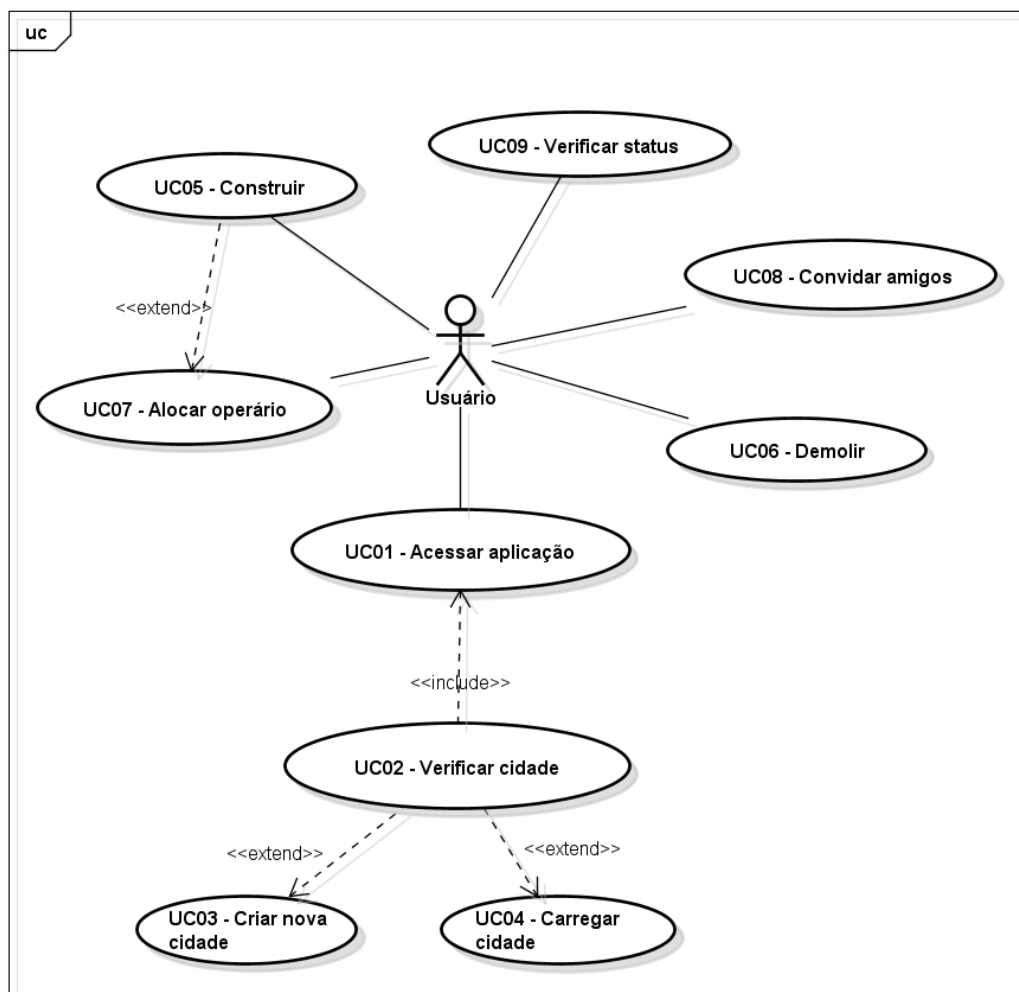


FIGURA 33: Diagrama de Casos de Uso para o jogo MyCity.

Apêndice 6 - Descrição dos Casos de Uso

Apêndice 6.1 UC01 - Acessar aplicação

Descrição

Este caso de uso mostra como ocorre o acesso do usuário.

Pré-condições

N/A

Pós-condições

Após o fim normal deste caso de uso o sistema deve:
Ter recuperado a identificação única do usuário.

Ator Primário

Usuário

Fluxo de Eventos Principal

1. Usuário loga-se no Facebook.
2. O usuário acessa o aplicativo “*playmycity*”.
3. O sistema recupera o identificador único do Facebook do usuário.
4. O sistema chama o caso de uso UC02 – Verificar cidade, passando o identificador como parâmetro.
5. O caso de uso é finalizado.

Apêndice 6.2 UC02 – Verificar cidade

Descrição

Este caso de uso verifica se já existe uma cidade criada para o usuário.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC01 - Acessar aplicação.

Pós-condições

N/A

Ator Primário

Sistema

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro: Identificador único do Facebook.
2. O sistema verifica na base de dados se existe uma cidade relacionada a este identificador.
3. O sistema não encontra uma cidade. (A1)
4. O sistema chama o caso de uso UC03 – Criar nova cidade, passando o identificador como parâmetro.
5. O caso de uso é finalizado.

Fluxo de Eventos Alternativo

A1 – O sistema encontra uma cidade

1. O sistema recupera a identificação da cidade.
2. O sistema chama o caso de uso UC04 – Carregar cidade, passando o identificador como parâmetro.
3. O caso de uso é finalizado.

Apêndice 6.3 UC03 - Criar nova cidade

Descrição

Este caso de uso mostra como deve ser criada uma nova cidade.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:
Ter criado um registro da cidade do usuário.

Ator Primário

Sistema

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro: Identificador único do Facebook.
2. O sistema recupera os dados da conta do Facebook do usuário. (R1)
3. O sistema insere um novo registro de uma cidade no banco de dados usando os dados obtidos do usuário e a configuração inicial de uma nova cidade.
4. O sistema apresenta ao usuário a interface inicial do jogo. (DV1)
5. O caso de uso é finalizado.

Regras

R1. Dados da conta do Facebook a serem recuperados:

- Nome
- Avatar

DV1 - Interface inicial do jogo



Apêndice 6.4 UC04 - Carregar cidade

Descrição

Este caso de uso mostra como deve ser recuperados os dados de uma cidade.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

Ter recuperado os dados da cidade e as suas construções existentes.

Ator Primário

Sistema

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro: Identificador único da cidade.
2. O sistema recupera do banco os dados da cidade usando o identificador na consulta.
3. O sistema recupera a lista de construções da cidade usando o identificador único da cidade na tabela relacional de construções da cidade.
4. O sistema passa os dados recuperados para o cliente do usuário.
5. O cliente atualiza a interface do usuário com suas construções e dados recuperados do banco. (DV1)
6. O caso de uso é finalizado.

Apêndice 6.5 UC05 - Construir

Descrição

Este caso de uso mostra como acontece o processo de construção de uma nova construção em uma cidade.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado **UC02 – Verificar cidade**

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

Ter adicionado os dados da nova construção da cidade do jogador.

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O usuário aciona a ação “*Construir*”.(DV1)
2. O sistema lista os tipos de construção para o usuário.(DV2)
3. O usuário seleciona um tipo de construção.
4. O sistema recupera as construções do tipo selecionado pelo usuário e os apresenta na interface do cliente.(DV3)
5. O usuário seleciona uma construção.
6. O sistema demonstra os dados da construção para o usuário.
7. O usuário aciona a ação “*Construir esta construção*”. (A1)(R1)
8. O cursor da interface é alterado para o preview da construção.
9. O usuário arrasta o preview da construção para uma posição desejada e clica no mapa. (A2)(A3)(R2)

10. O sistema grava a construção na tabela relacional de construções da cidade usando a localização do mapa.
11. O sistema atualiza os operários da cidade.
12. O sistema atualiza a interface do usuário com a animação do processo de construção, e coloca um contador com o tempo restante da conclusão. **(A4)**
13. Após o tempo de construção terminar o sistema atualiza os atributos da cidade com os valores de bônus da construção criada.
14. O caso de uso é finalizado.

Fluxo de Eventos Alternativos

A1 – Os requerimentos para construção não foram atendidos

1. O sistema apresenta uma mensagem com a explicação do motivo da construção não poder ser construída. **(DV4)**
2. O caso de uso é finalizado.

A2 – O usuário clica numa posição já ocupada por outra construção.

1. O preview da construção fica avermelhado e o usuário não pode selecionar esta posição.
2. O fluxo é alterado para o **fluxo 9 do evento principal**.

A3 – O usuário pressiona ESC.

1. O caso de uso é finalizado.

A4 – O usuário aciona a ação “Adicionar construtor”.

1. O sistema chama o caso de uso **UC07 - Alocar operário**.
2. Após o final do caso de uso chamado o sistema volta ao **fluxo 12 do evento principal**.

Regras

R1 – Validação de construção disponível

O sistema verifica quais são as dependências da construção selecionada pelo usuário. As dependências próprias da construção são: custo em dinheiro, custo em crédito e população mínima. Existem as dependências de outras construções, onde deve existir um mínimo de construções já presentes na cidade para poder ser válida.

R2 – Validação de posição a ser construída uma construção

O sistema verifica se existe uma parte de uma construção em qualquer posição que o usuário posicionou o preview da construção. Se houver um quadro ocupado a interface bloqueia a construção neste local.

DV1 - Ação Construir



DV2 - Tipos de construções



DV3 - Construções do tipo selecionado



DV4 - Erro ao selecionar construção



Apêndice 6.6 UC06 - Demolir

Descrição

Este caso de uso serve para demolir uma construção da cidade.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

Ter removido uma construção da cidade do jogador.

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O usuário aciona a ação “*Demolir*”. (DV1)
2. O usuário seleciona a construção a ser demolida com um clique. (DV1)
3. O sistema valida se a construção pode ser demolida. (R1)
4. O sistema apresenta um pop-up pedindo a confirmação da demolição. (A1)
5. O usuário confirma a demolição. (A2)
6. O sistema remove do banco de dados a construção selecionada da cidade do usuário.
7. O sistema atualiza a interface do usuário com a animação do processo de demolição.
8. O caso de uso é finalizado.

Fluxo de Eventos Alternativos

A1 – A construção não pode ser removida

1. O sistema apresenta uma mensagem com a explicação do motivo da construção não poder ser removida.
2. O caso de uso é finalizado.

A2 – O usuário não aceita a demolição

1. O caso de uso é finalizado.

Regras

R1 – Validação de demolição

O sistema verifica se o campo removível desta construção é verdadeiro ou falso. Se for falso o processo de demolição é interrompido e o usuário é informado. Se for verdadeiro, o sistema deve verificar se esta construção é um pré-requisito de alguma outra construção da cidade, e somente não havendo nenhuma dependência com as outras construções a construção pode ser removida.

DV1 - Ação para remover construção



Apêndice 6.7 UC07 - Alocar operário

Descrição

Este caso de uso serve para alocar operários numa construção em andamento.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC05 – Construir ou tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

-

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O usuário aciona a ação “Adicionar construtor”.(DV1)
2. O sistema valida a possibilidade de alocar um operário. (E1)(E2)(R1)
3. O usuário seleciona a quantidade de operários e confirma a alocação.
4. (DV1)
5. O sistema atualiza o número de operários da construção em andamento.
6. O sistema atualiza o número de operários disponíveis da cidade.
7. Na interface, o contador regressivo da construção é atualizado. (R2)
8. O fluxo é desviado para a etapa 12 do caso de uso UC05 – Construir.
9. O caso de uso é finalizado.

Fluxo de Eventos de Exceção

E1 – A cidade não possui operários disponíveis.

1. O sistema apresenta a mensagem: "Não há operários disponíveis na cidade".
 2. O caso de uso é finalizado.
- E2 – A construção já possui o máximo de operários.
1. O sistema apresenta a mensagem: "Não podem ser alocados novos operários, pois a capacidade máxima já foi alcançada".
 2. O caso de uso é finalizado.

Regras

R1 – Validação de alocação de operário

O sistema verifica a quantidade alocada de construtores na construção com a quantidade máxima de operários da construção. Somente se a quantidade alocada for menor que a máxima o sistema irá verificar se a cidade possui operários disponíveis para a alocação de um operário.

R2 – Atualização de tempo de construção

O sistema verifica a quantidade alocada de construtores na construção e recupera o tempo restante da construção e atualiza o tempo restante utilizando esta fórmula:

$$TR_{rf} = TR_{rc} * 0,8OP - 1$$

Sendo:

TR_f = tempo restante final

TR_c = tempo restante da construção

OP = número de operários a serem alocados + número de operários já alocados

DV1 - Alocação de operários



Apêndice 6.8 UC08 - Convidar amigos

Descrição

Este caso de uso serve para convidar amigos do Facebook do usuário a jogar

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

-

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O usuário aciona a ação “*Convidar amigos*”.(DV1)
2. O sistema recupera a lista de amizades do usuário usando sua identificação única do Facebook, sendo que apenas recuperará aqueles que ainda não jogam. (E1)
3. O sistema apresenta uma lista com os nomes e imagens dos amigos.(DV1)
4. O usuário seleciona os amigos que deseja convidar.(DV1)
5. O usuário preenche uma mensagem para o convite.
6. O usuário clica em “Convidar estes amigos”.
7. O sistema salva na tabela de convites a identificação dos amigos que foram convidados e de quem fez o convite.
8. O sistema faz uma requisição ao Facebook para convite da aplicação.
9. O sistema apresenta a mensagem “Os convites foram enviados”.
10. O caso de uso é finalizado.

Fluxo de Eventos de Exceção

E1 – O usuário não possui amigos para convidar

1. O sistema apresenta a mensagem “Você não possui amigos a convidar”
2. O caso de uso é finalizado.

DV1 - Convidar amigos



Apêndice 6.9 UC09 - Verificar status

Descrição

Este caso de uso serve para visualizar os status da cidade e das construções.

Pré-condições

Este caso de uso pode iniciar somente se:

O sistema tiver executado UC02 – Verificar cidade

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

-

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O usuário posiciona o cursor do mouse sobre uma construção.(A1)(A2)(DV1)
2. O sistema recupera as informações sobre a construção. (R1)
3. O sistema atualiza a interface, posicionando um pop-up sobre a construção selecionada com os dados recuperados.(DV1)
4. O usuário remove o cursor do mouse da construção.(A3)
5. O caso de uso é finalizado.

Fluxo de Eventos Alternativos

A1 – O usuário clica em *Visualizar mais dados*.(DV1)

1. O sistema recupera os dados da cidade. (R2)
2. O sistema atualiza a interface, criando um pop-up com os dados recuperados.
3. O usuário clica no botão *Fechar*.(A3)
4. O caso de uso é finalizado.

A2 – O usuário clica em *Estatísticas*. (DV1)

1. O sistema recupera os dados da cidade e das construções desta. (R3)
2. O sistema atualiza a interface, criando um pop-up com os dados recuperados.
3. O usuário clica no botão *Fechar*. (A3)
4. O caso de uso é finalizado.

Regras

R1 – Dados a serem recuperados das construções.

O sistema recupera os dados conforme o tipo de construção. Tipos residenciais terão dados sobre a população suportada/alocada, tipos de comércio terão dados sobre empregos e renda mensal, tipos públicos terão dados sobre cobertura das necessidades da população (Ex.: número de médicos por habitante).

R2 – Dados a serem recuperados da cidade.

O sistema recupera os dados dos indicadores da cidade, população atual, renda, impostos cobrados e despesas públicas.

R3 – Dados a serem recuperados para as estatísticas.

O sistema compara os indicadores da cidade, bem como a renda total com as construções da cidade e faz uma relação entre eles e a população, apresentando dados como crescimento de empregos, melhoramento do IDH e demais mudanças na cidade.

DV1 - Alocação de operários



Apêndice 7 - Diagrama Entidade Relacionamento

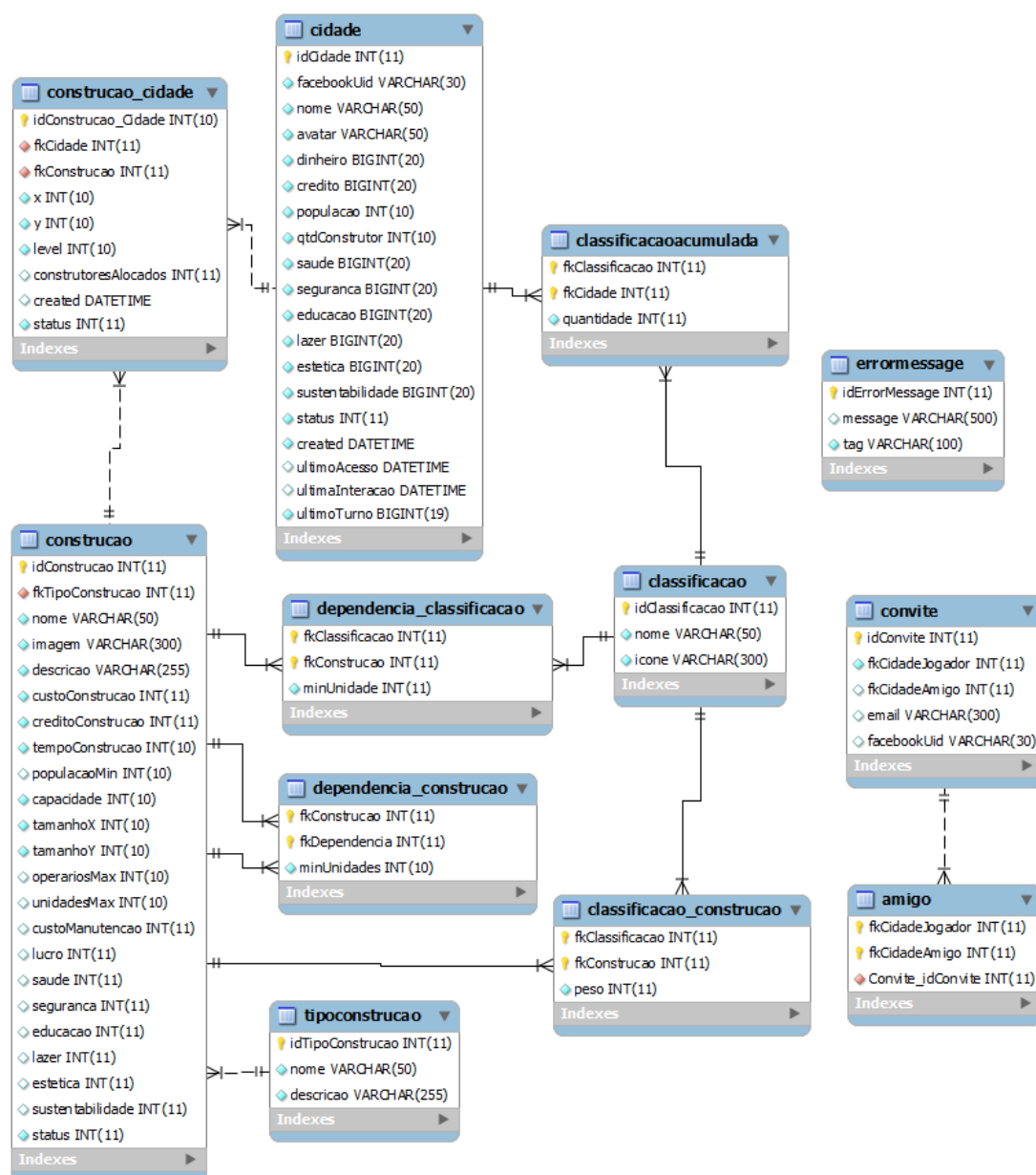


FIGURA 34:
MyCity.

Diagrama do Modelo de Entidade Relacionamento para o jogo

Apêndice 8 - Análise por Pontos por Caso de Uso

Tipo de Ator	Peso			
Ator Simples	1			
Ator Médio	2			
Ator Complexo	3	Total		
		1	3	
	Peso por atores (UAW)	3		
Tipo de Caso de Uso	Número de Transações	Peso	Qtd.	Total
Simples	Até 3	1	4	4
Médio	4 a 7	2	3	6
Complexo	7 ou mais	3	2	6
	Peso por casos de uso(UUCW)	16		
Fator	Requisito	Peso	Qtd.	Total
T1	Sistema distribuído	2	2	4
T2	Tempo de Resposta	2	4	8
T3	Eficiência	1	5	5
T4	Processamento complexo	1	3	3
T5	Código reusável	2	4	8
T6	Facilidade de instalação	1	1	1
T7	Facilidade de uso	0,5	2	1
T8	Portabilidade	2	1	2
T9	Facilidade de mudança	1	3	3
T10	Concorrência	1	1	1
T11	Recursos de segurança	1	5	5
T12	Acessível por terceiros	1	5	5
T13	Requer treinamento especial	1	1	1
	Pontos por complexidade técnica(TF)	47		
TCF = $0.6 + (0.01 \times \text{TFactor})$	Fator de complexidade técnica	1,07		
Fator	Requisito	Peso	Qtd.	Total
E1	Familiaridade com RUP ou outro processo formal	1,5	4	6
E2	Experiência com a Aplicação em desenvolvimento	0,5	5	2,5
E3	Experiência em Orientação a Objetos	1	5	5
E4	Presença de analista experiente	0,5	2	1
E5	Motivação	1	3	3
E6	Requisitos estáveis	2	4	8
E7	Desenvolvedores em meio-expediente	-1	3	-3
E8	Linguagem de programação difícil	2	2	4
	Pontos por fatores ambientais(TE)	26,5		
EF = $1.4 + (-0.03 \times \text{EFactor})$	Fator ambiental	0,605		
UUCP = UAW + UUCW	Pontos por caso de uso não ajustado	19		
UCP = UUCP x TCF x EF	Pontos por caso de uso	12,29965		
Tempo = 12 (UCP) *20 = 240 horas.	Utilizando 20 homens hora temos um tempo total de	245,993	horas	

FIGURA 35:

Análise por Pontos por Caso de Uso.

Apêndice 9 - Diagramas de Atividades

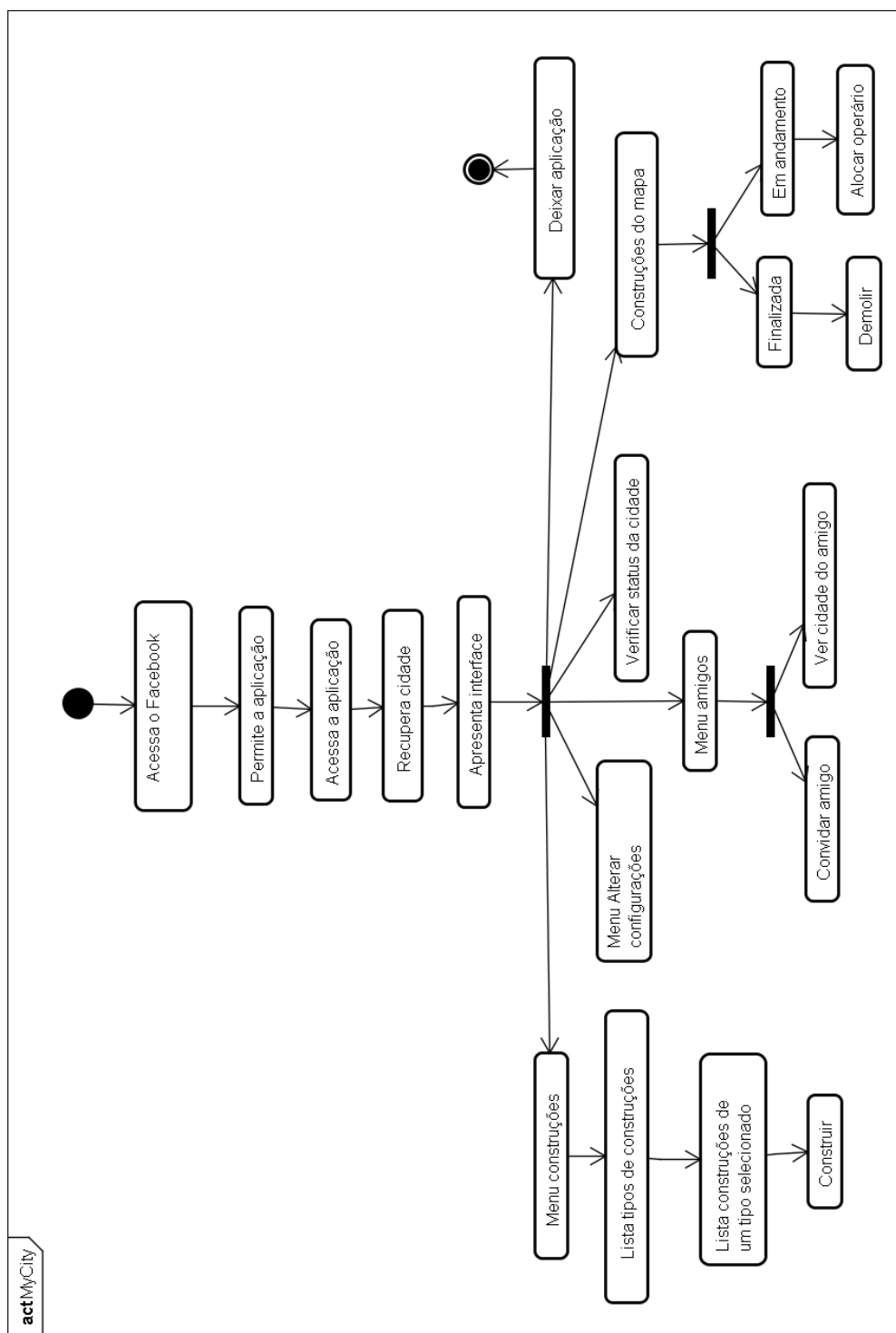


FIGURA 36: Diagrama de atividades para acessar a aplicação.

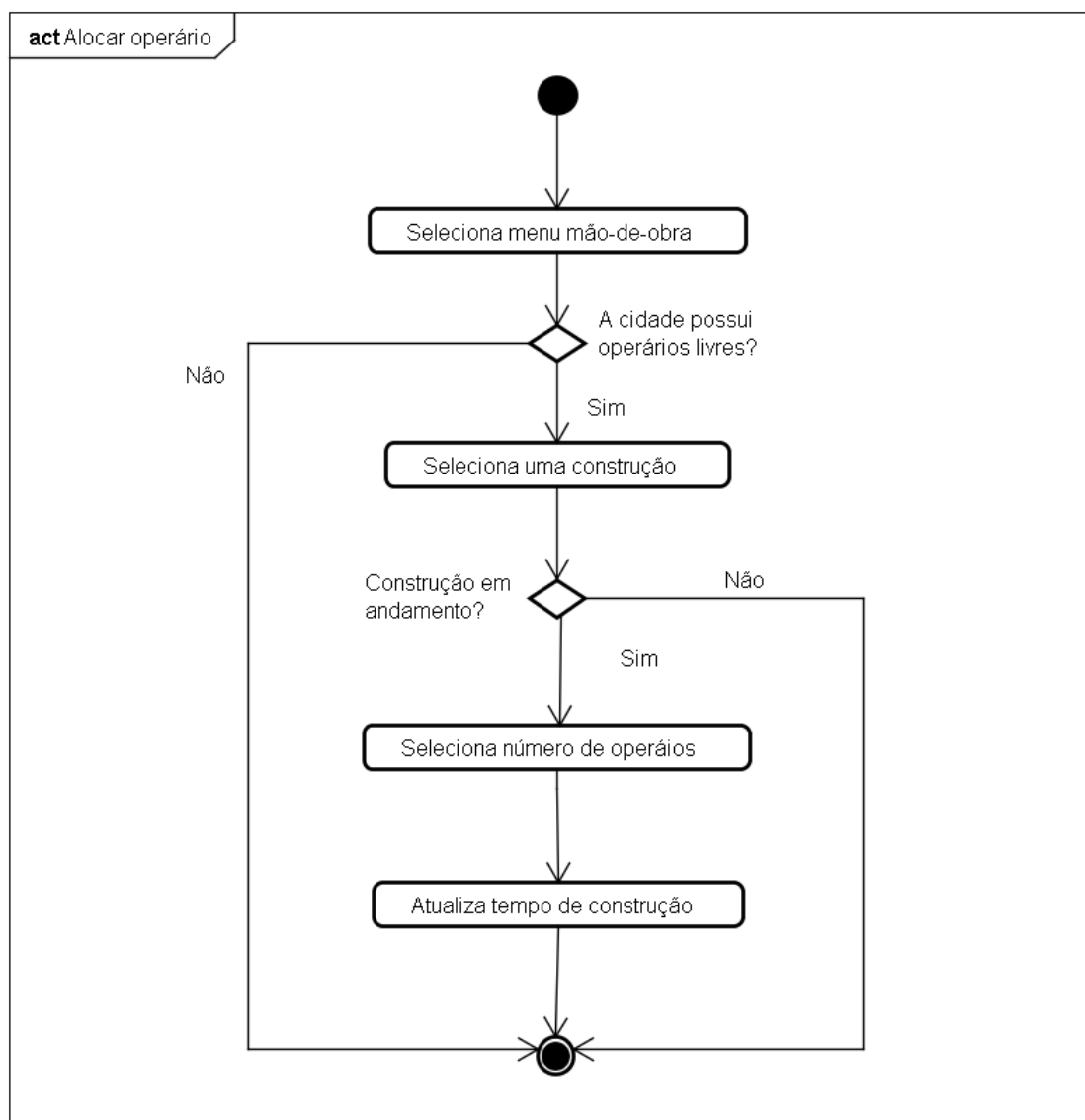


FIGURA 37: Diagrama de atividades para alocar um operário.

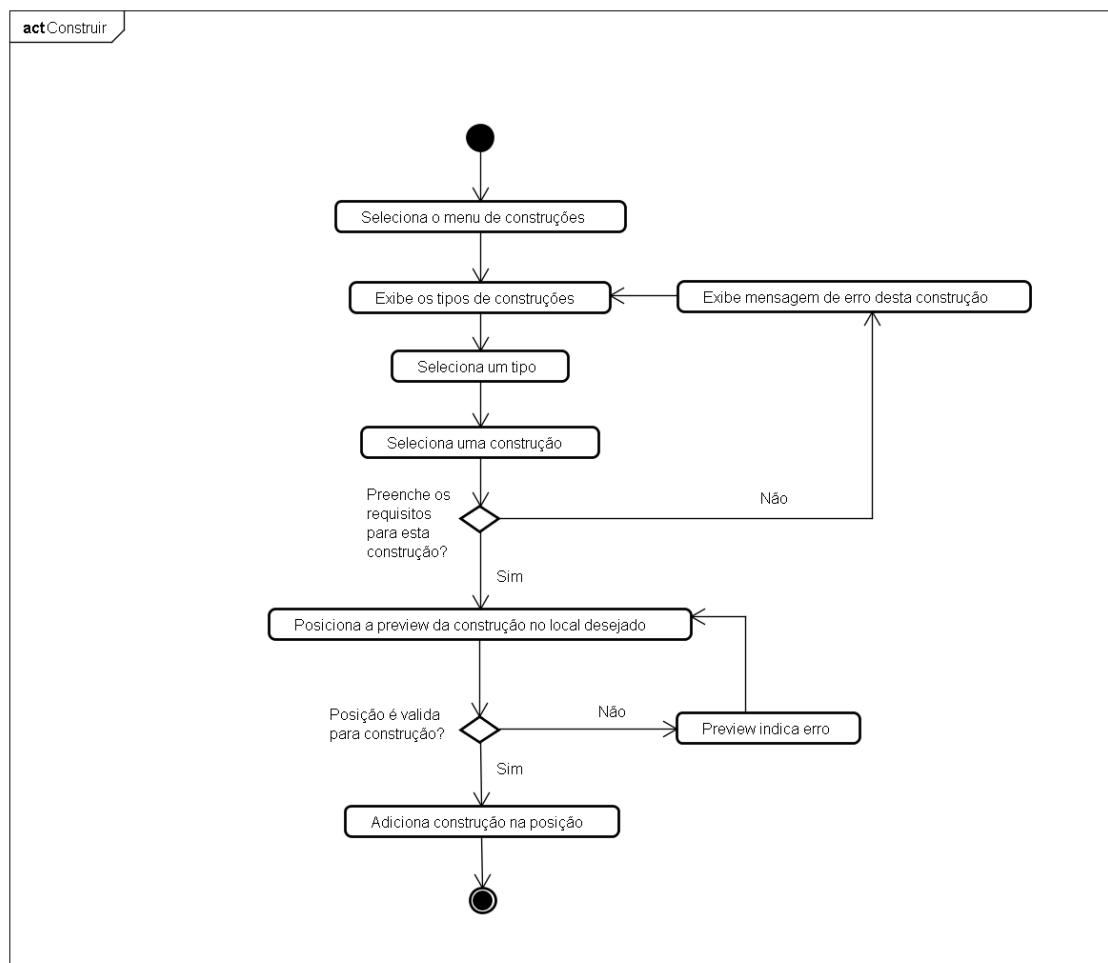


FIGURA 38: Diagrama de atividades para construir um item.

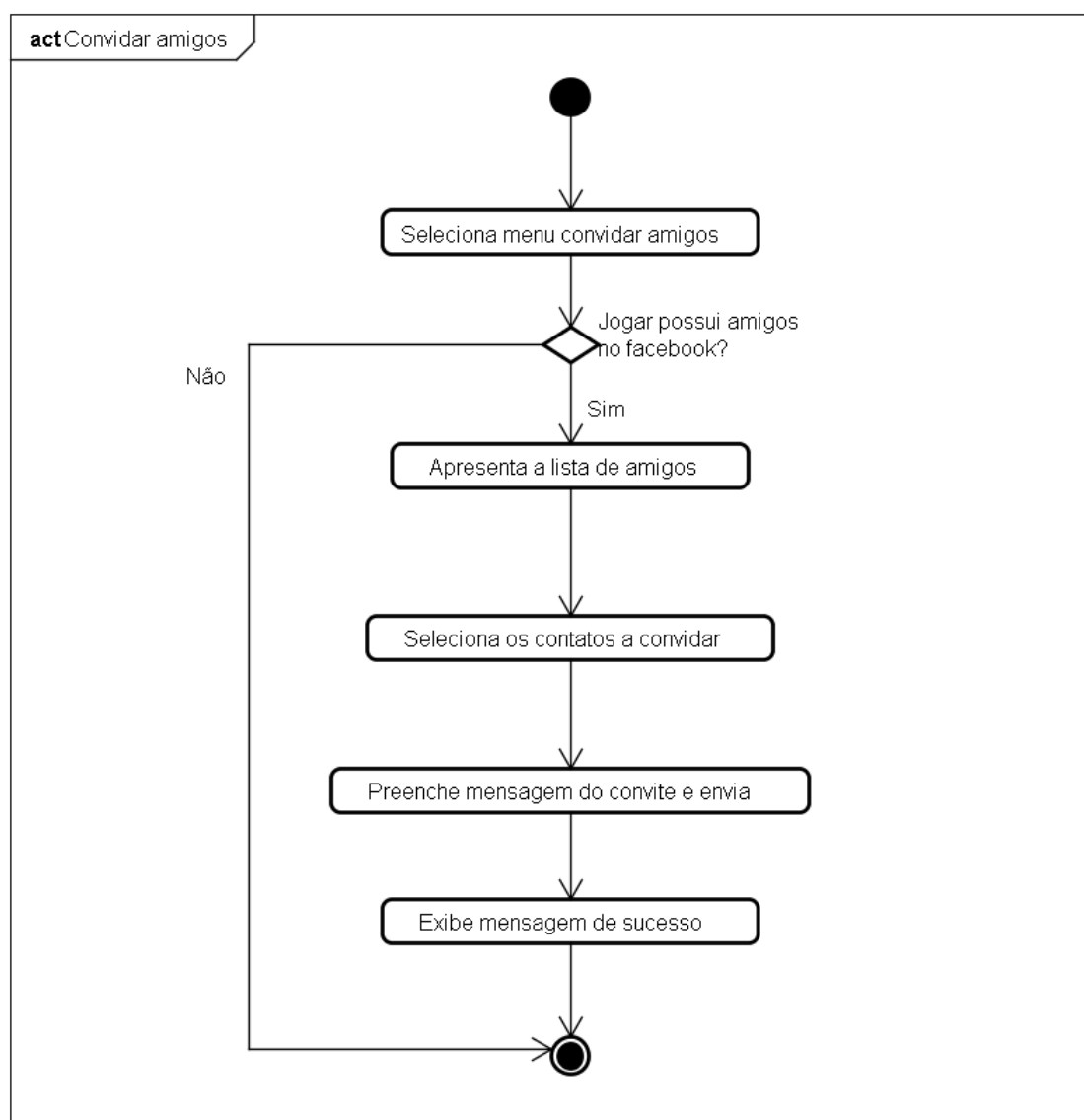


FIGURA 39: Diagrama de atividades para convidar amigos do Facebook.

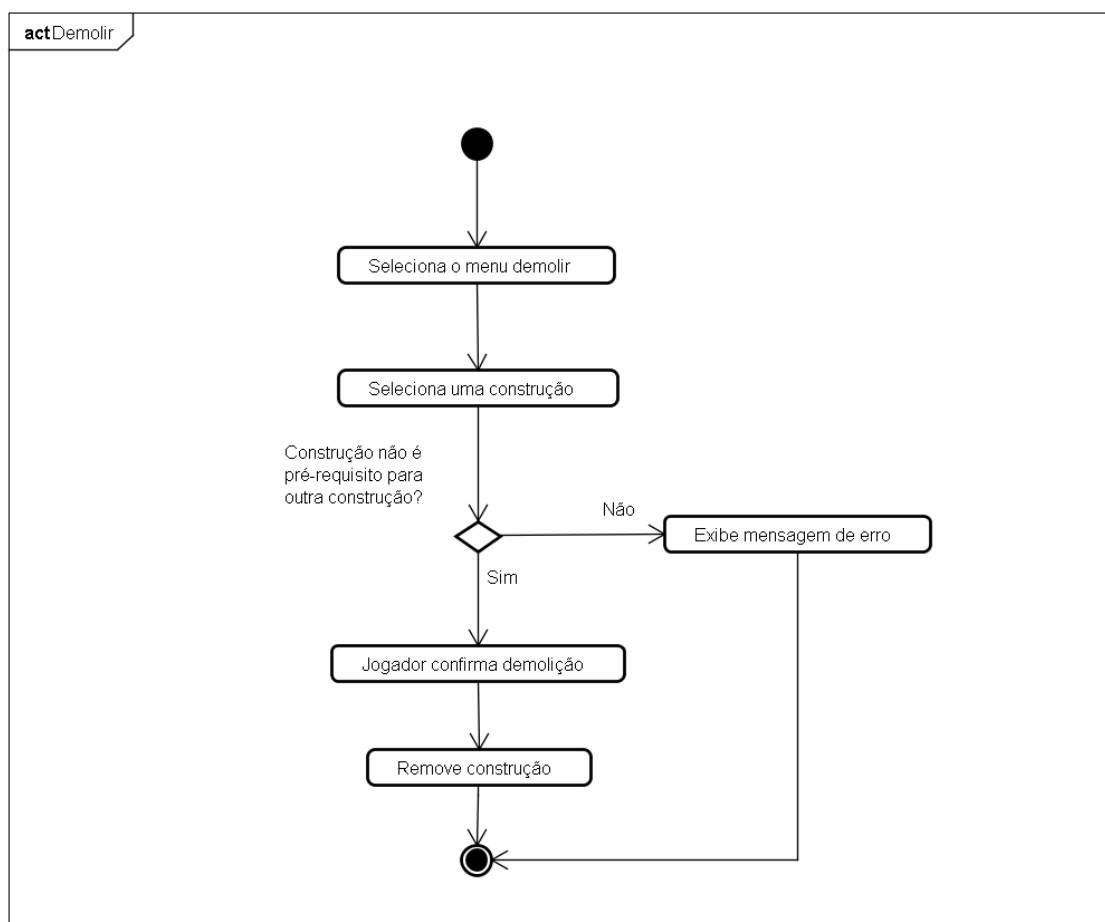


FIGURA 40: Diagrama de atividades para demolir uma construção.

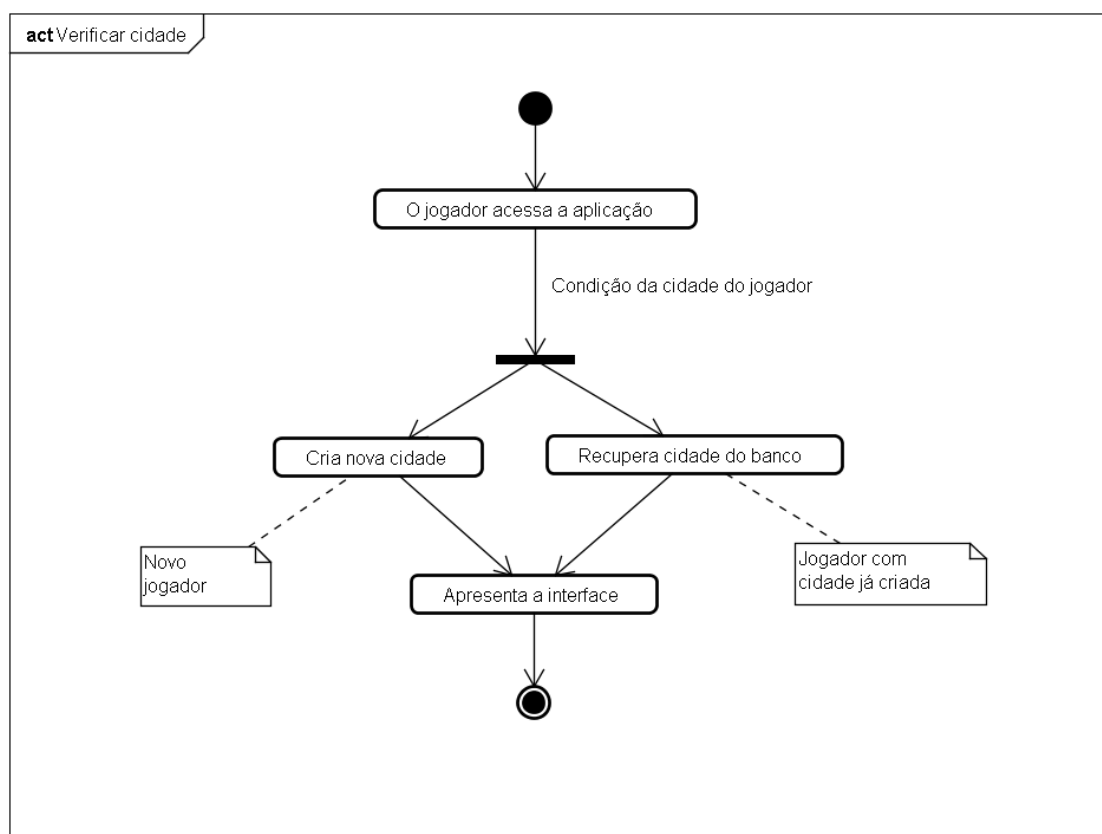


FIGURA 41:
jogador.

Diagrama de atividades para verificar a cidade durante o acesso do

Apêndice 10 - Diagrama de Estados

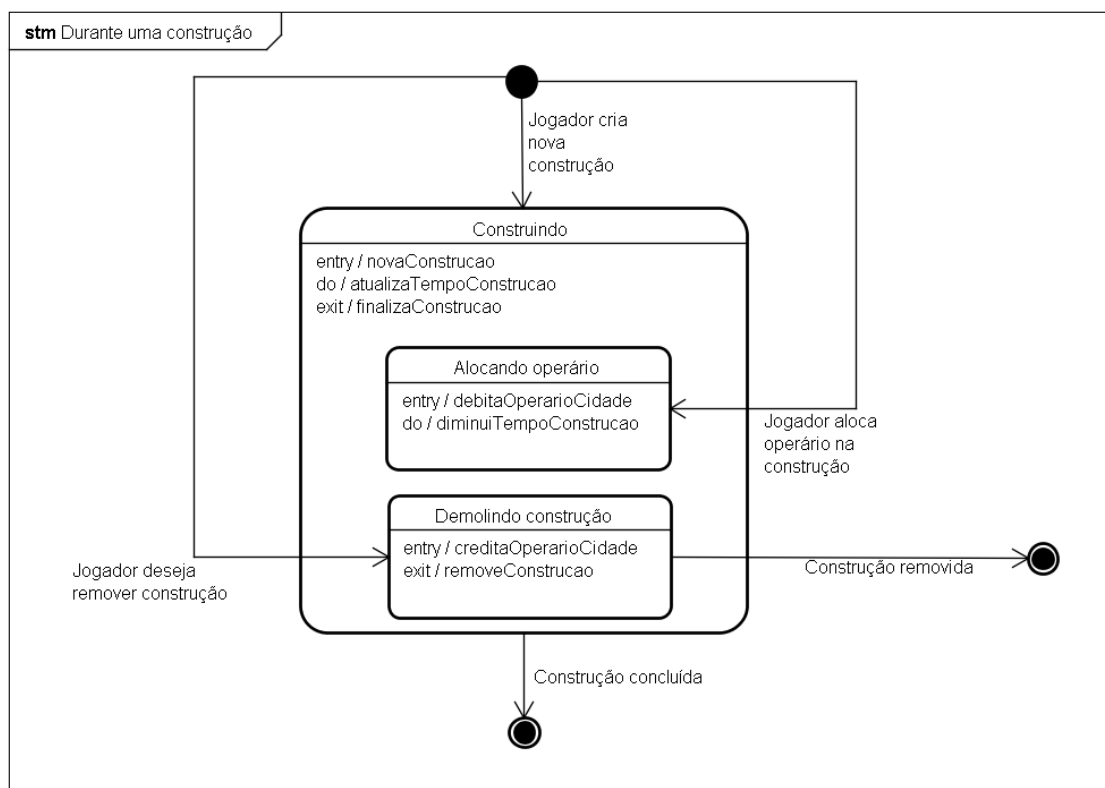


FIGURA 42: Diagrama de estado que ocorre durante uma construção.

Apêndice 11 - Diagrama de Sequencia

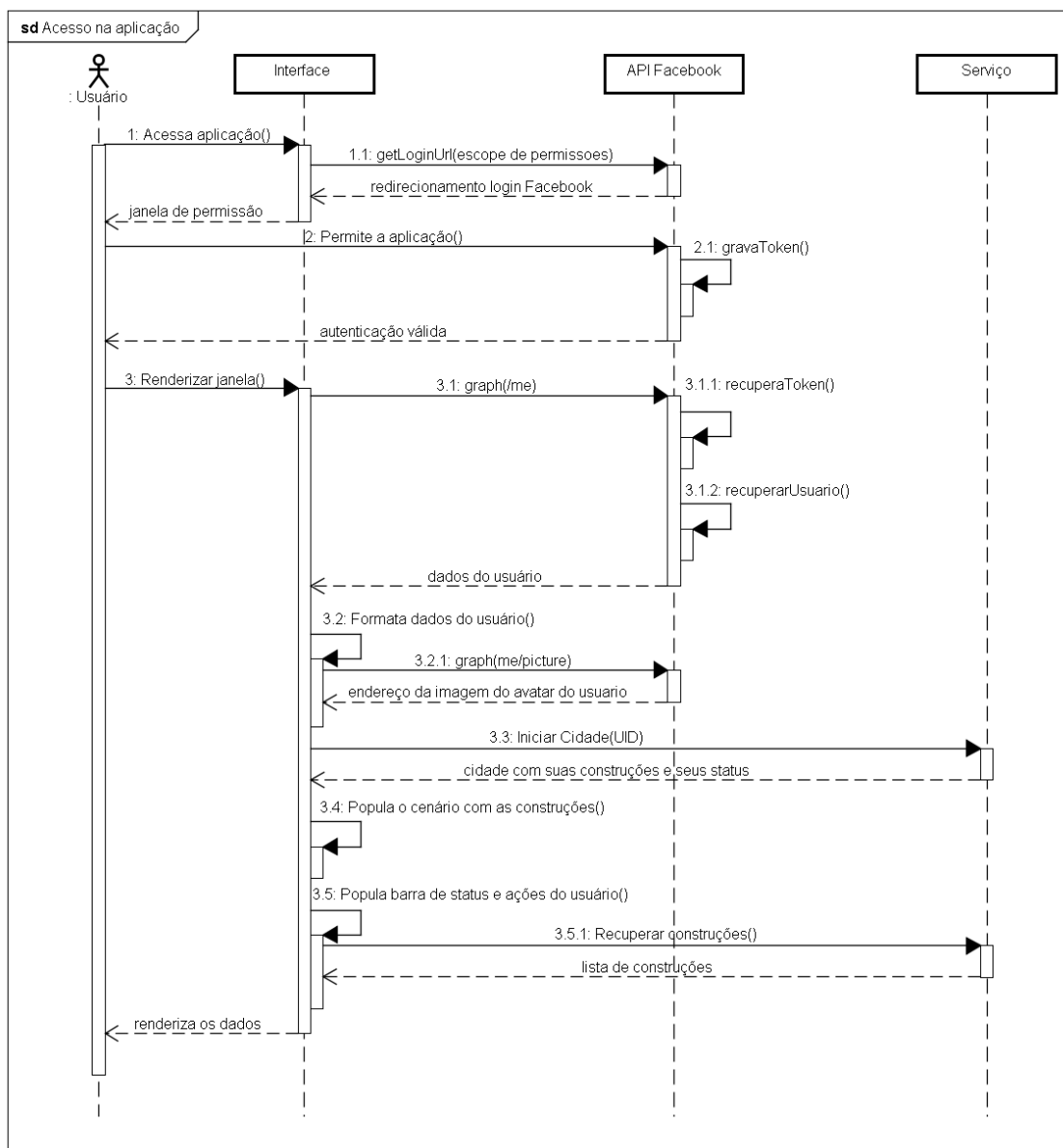


FIGURA 43: Diagrama descrevendo o acesso à aplicação

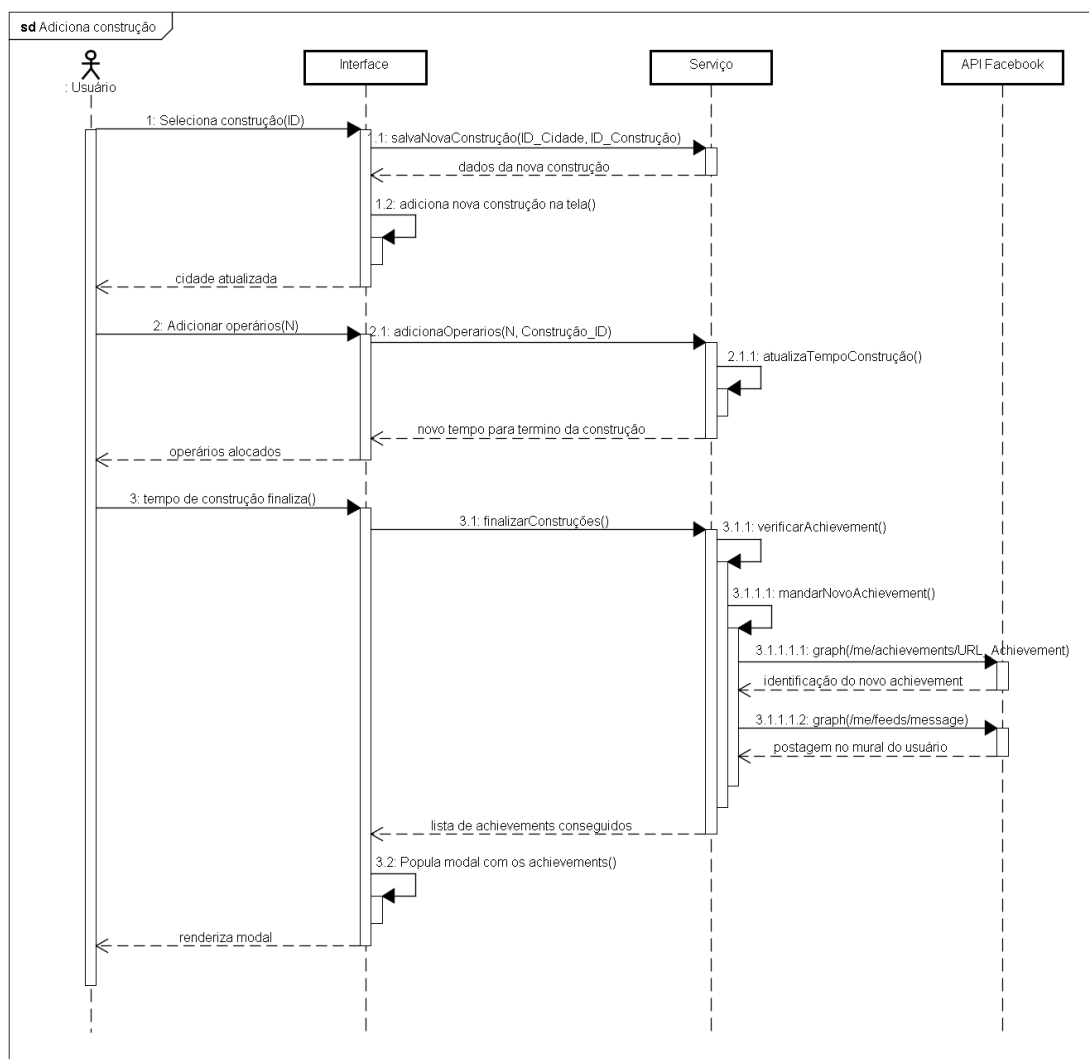


FIGURA 44: Diagrama ilustrando o processo para se adicionar uma construção

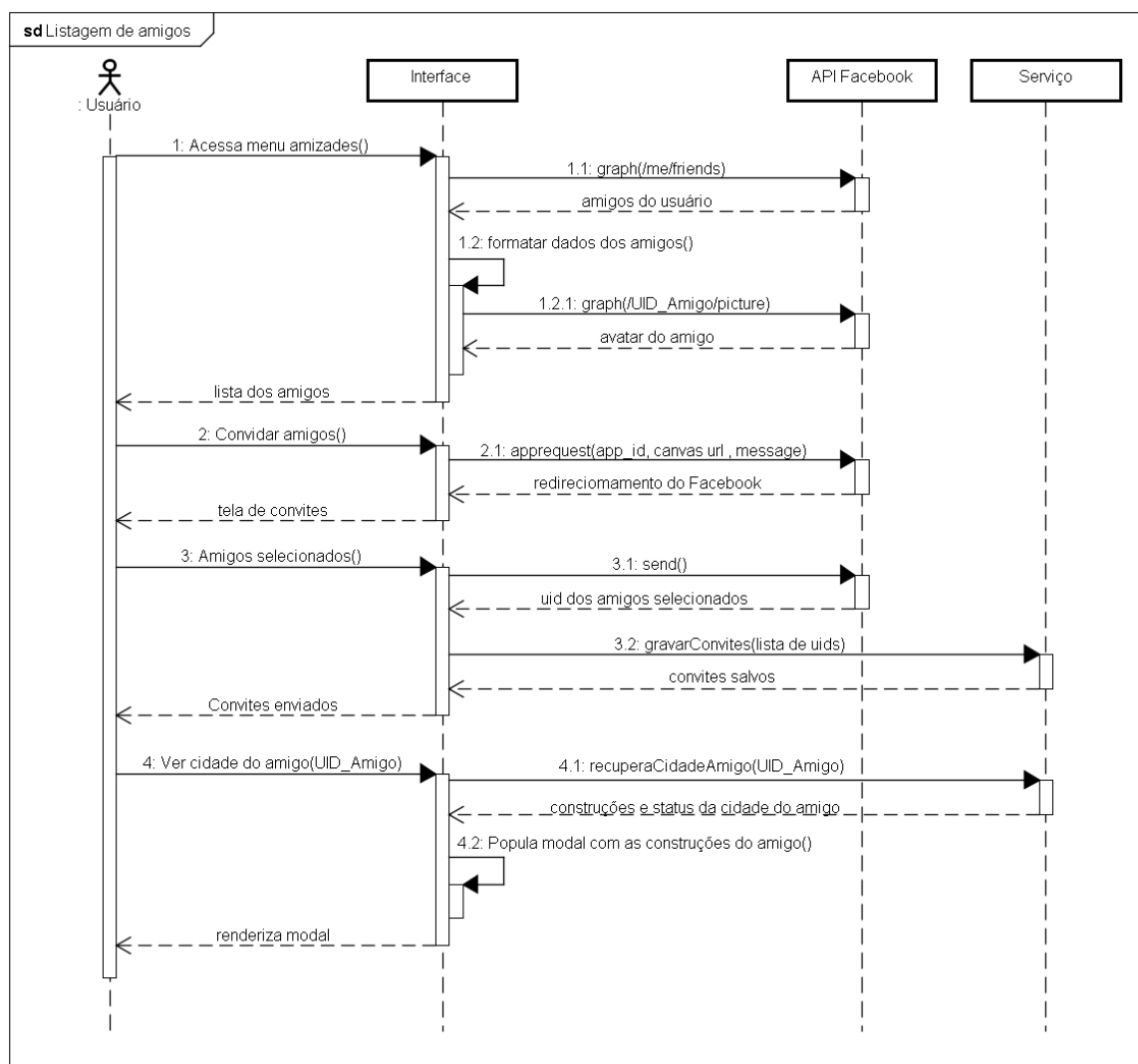


FIGURA 45:
amigos

Diagrama de sequencia descrevendo o processo de listagem de